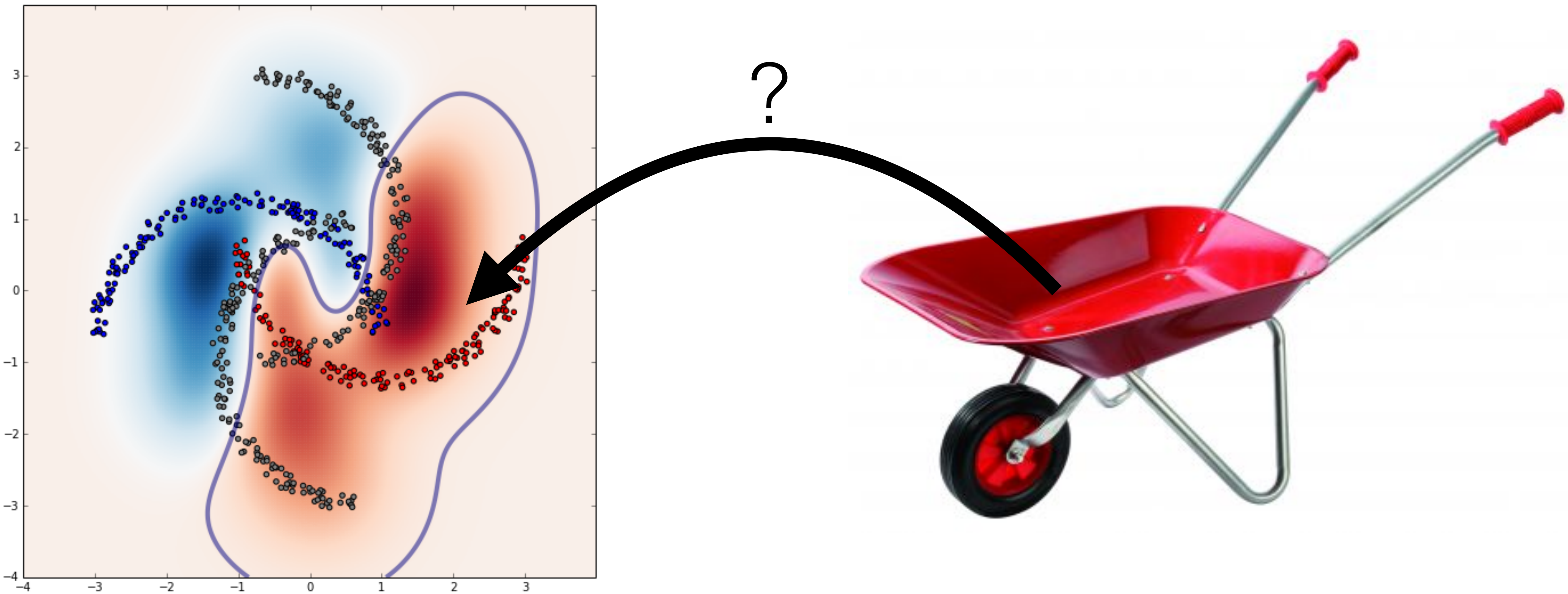


Transport optimal et Science des données

Nicolas Courty
UBS / IRISA / INRIA - Equipes Panama et Obélix
Meetup ML - Rennes - Juin 2019

Disclaimer: no Civil Engineering at all during the presentation

1. What is optimal transport ?
2. How can it be used in data science ?

1. What is optimal transport ?
2. How can it be used in data science ?

What is Optimal Transport ?

The natural geometry for probability measures

Monge Kantorovich Koopmans Dantzig Brenier Otto McCann Villani

Nobel ’75 Fields ’10

Origins: Monge Problem (1781)

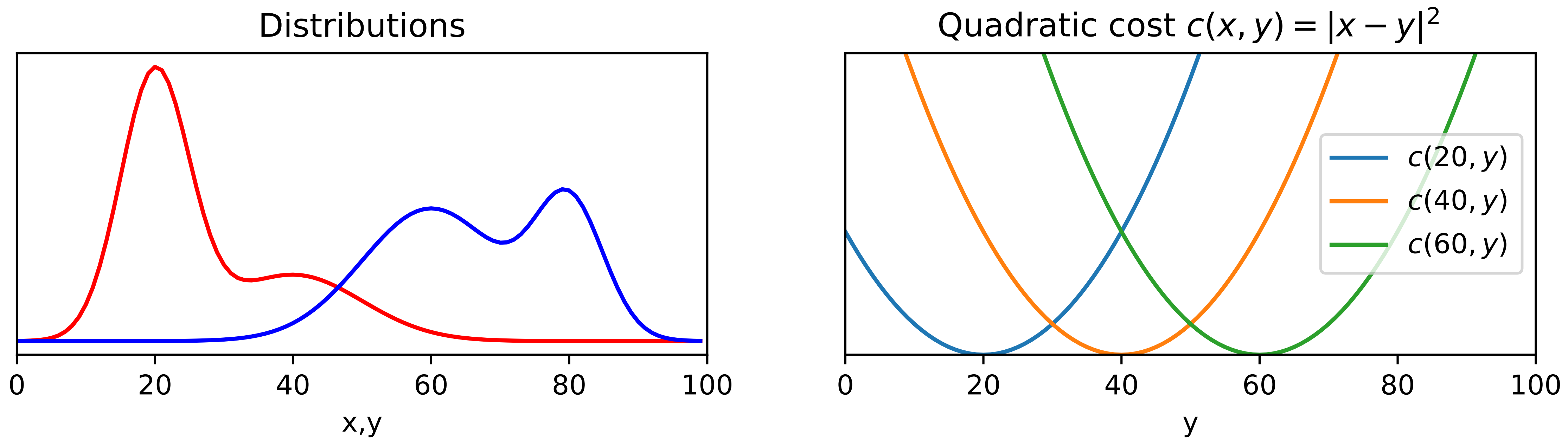
60 MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE SUR LA THÉORIE DES DÉBLAIS ET DES REMBLAIS.

Par M. MONGE.

LORSQU'ON doit tranſporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit tranſporter, & le nom de *Remblai* à l'eſpace qu'elles doivent occuper après le tranſport.

Optimal transport (Monge formulation)



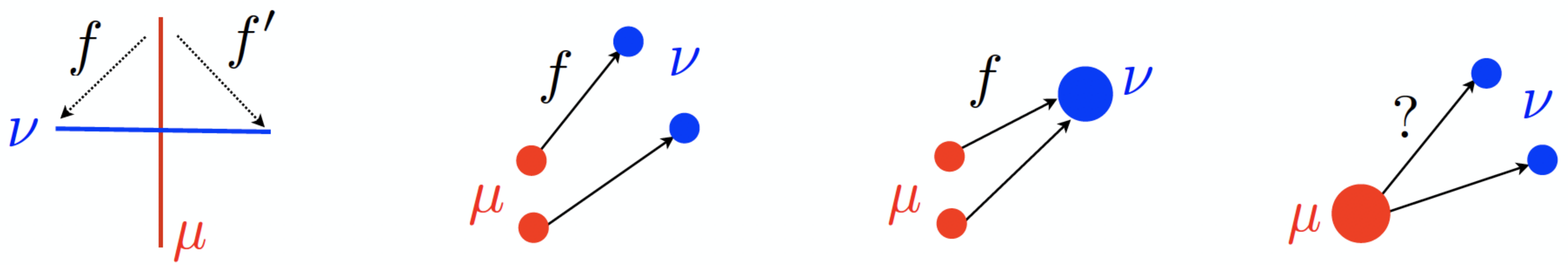
- Probability measures μ_s and μ_t on and a cost function $c : \Omega_s \times \Omega_t \rightarrow \mathbb{R}^+$.
- The Monge formulation [Monge, 1781] aim at finding a mapping $T : \Omega_s \rightarrow \Omega_t$

$$\inf_{T\#\mu_s=\mu_t} \quad \int_{\Omega_s} c(\mathbf{x}, T(\mathbf{x}))\mu_s(\mathbf{x})d\mathbf{x} \tag{1}$$

Non-existence / Non-uniqueness

Solving for this push-forward operator is a non-convex optimization problem,

- for which existence is not guaranteed,
- nor unicity



Note: [Brenier, 1991] proved existence and unicity of the Monge map for $c(x, y) = \|x - y\|^2$ and distributions with densities (i.e. continuous).

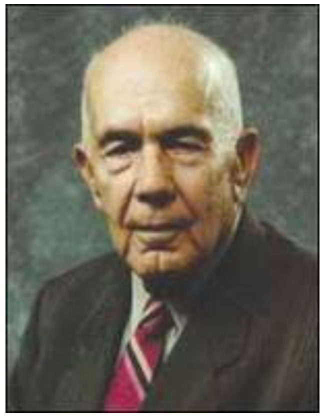
Kantorovich Problem

Kantorovich

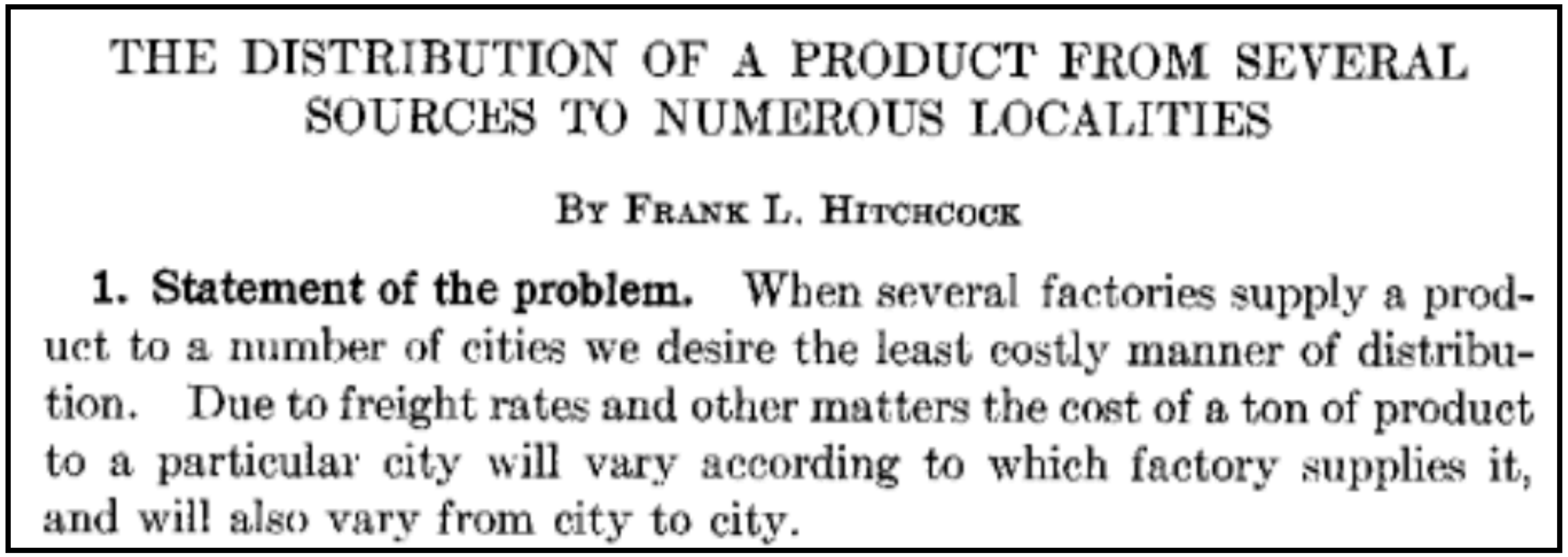
1939

Tolstoi

1930



Hitchcock



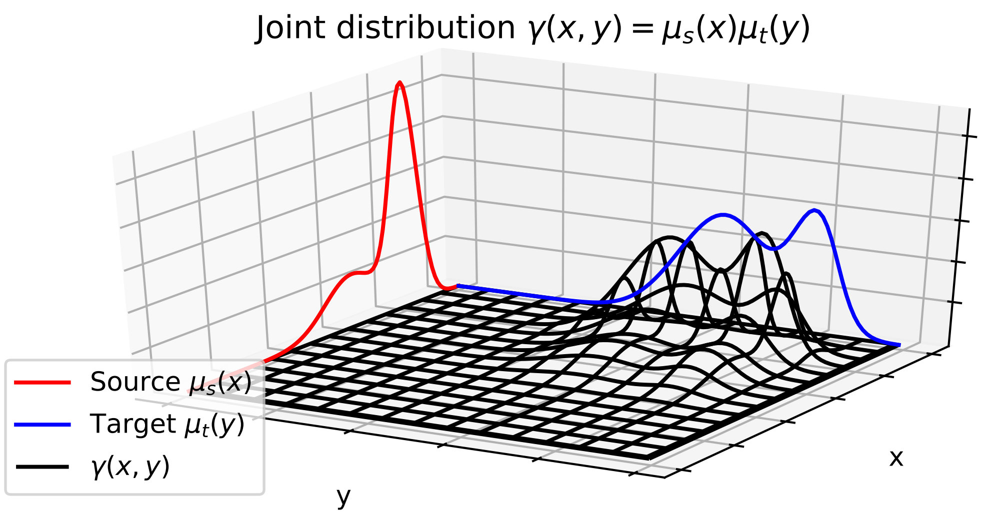
THE DISTRIBUTION OF A PRODUCT FROM SEVERAL SOURCES TO NUMEROUS LOCALITIES

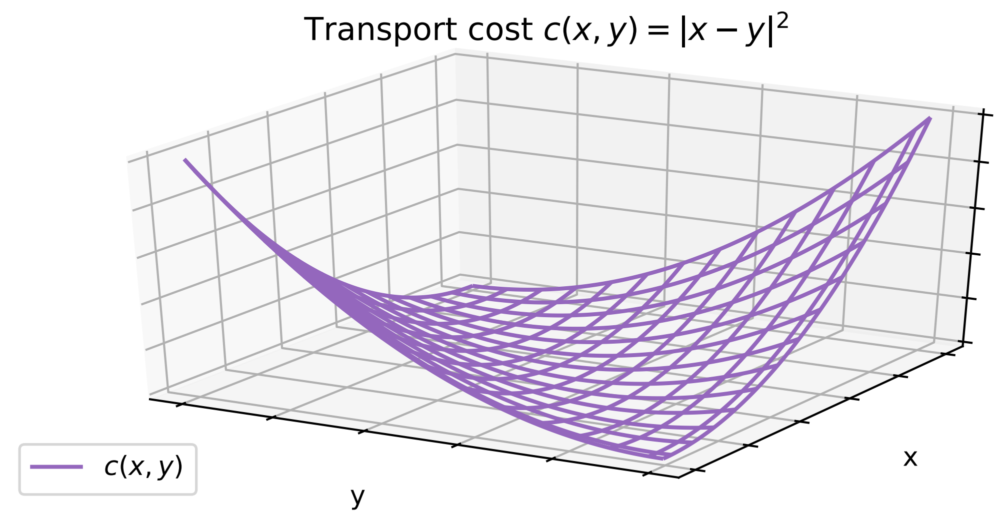
By Frank L. Hitchcock

1. Statement of the problem. When several factories supply a product to a number of cities we desire the least costly manner of distribution. Due to freight rates and other matters the cost of a ton of product to a particular city will vary according to which factory supplies it, and will also vary from city to city.

1941

Optimal transport (Kantorovich formulation)





- The Kantorovich formulation [Kantorovich, 1942] seeks for a probabilistic coupling $\gamma \in \mathcal{P}(\Omega_s \times \Omega_t)$ between Ω_s and Ω_t:

$$\boldsymbol{\gamma}_0 = \underset{\gamma}{\operatorname{argmin}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y})\gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}, \tag{2}$$

$$\text{s.t.} \quad \gamma \in \mathcal{P} = \left\{ \gamma \geq \mathbf{0}, \int_{\mathbf{\Omega_t}} \gamma(\mathbf{x}, \mathbf{y}) \mathbf{dy} = \boldsymbol{\mu_s}, \int_{\mathbf{\Omega_s}} \gamma(\mathbf{x}, \mathbf{y}) \mathbf{dx} = \boldsymbol{\mu_t} \right\}$$

- γ is a joint probability measure with marginals μ_s and μ_t.
- Linear Program that always have a solution.

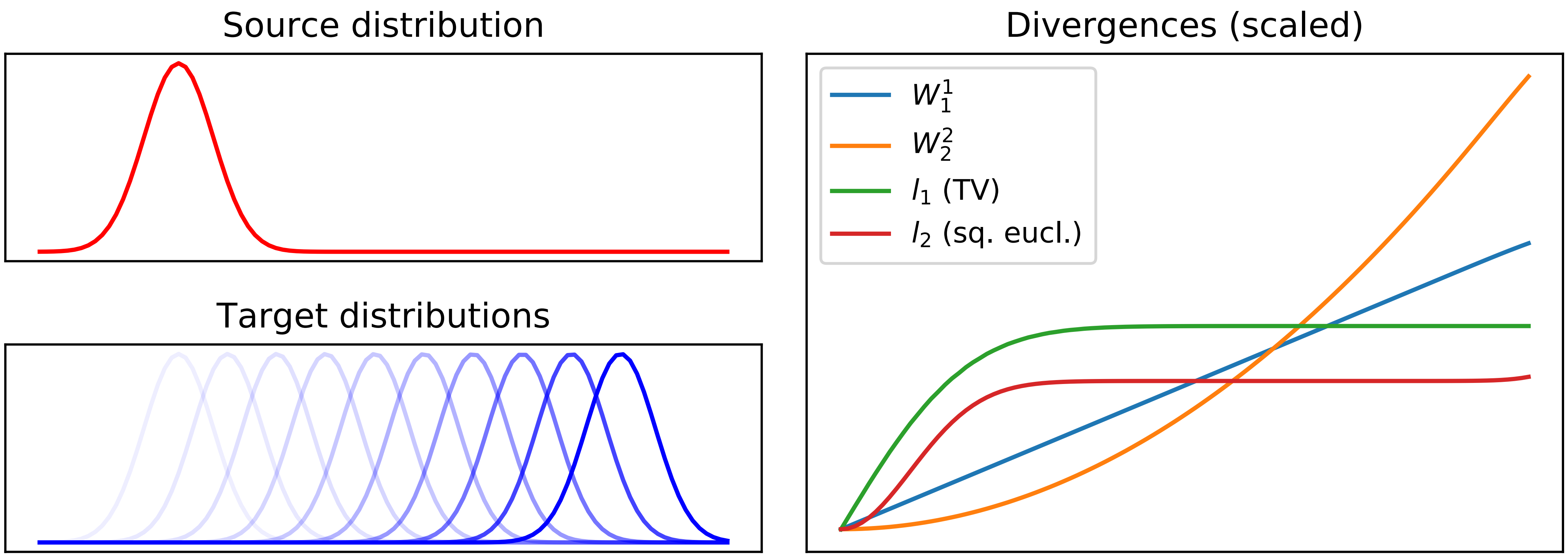
A first simple example

Matching words embedding

- Words are embedded in a high-dimensional space with neural networks
- Matching two documents is an OT problem, with the cost being the l_2 distance in the embedded space

Wasserstein distance

Wasserstein distance



Wasserstein distance

$$W_p^p(\mu_s, \mu_t) = \min_{\gamma \in \mathcal{P}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y}) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = \mathop{\mathbb{E}}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} [c(\mathbf{x}, \mathbf{y})] \tag{3}$$

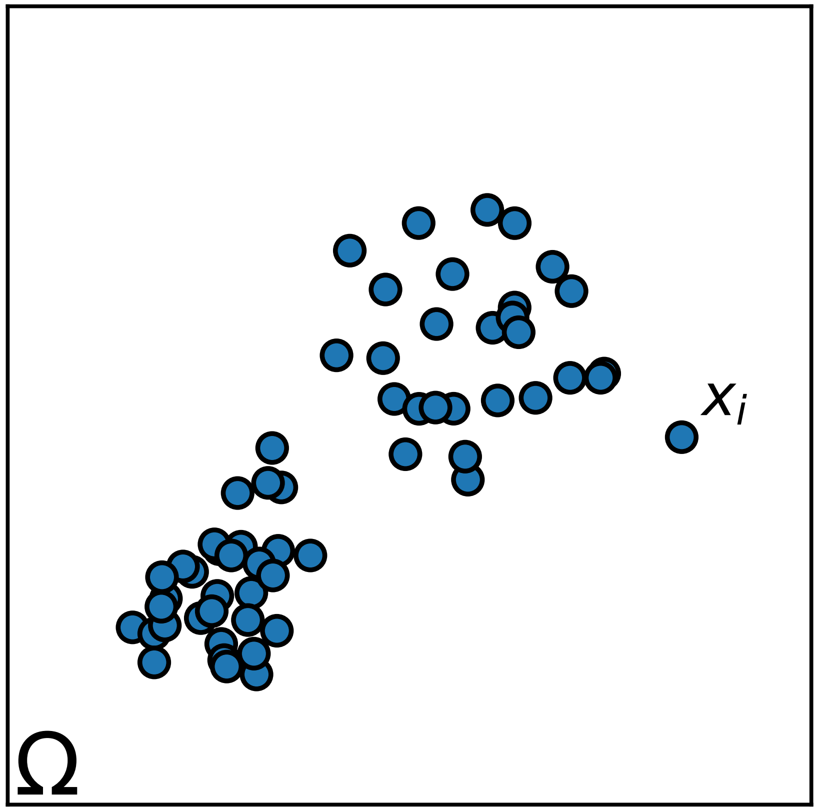
where $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$

- A.K.A. Earth Mover's Distance (W_1^1) [Rubner et al., 2000].
- Do not need the distribution to have overlapping support.
- Works for continuous and discrete distributions (histograms, empirical).

Discrete distributions: Empirical vs Histogram

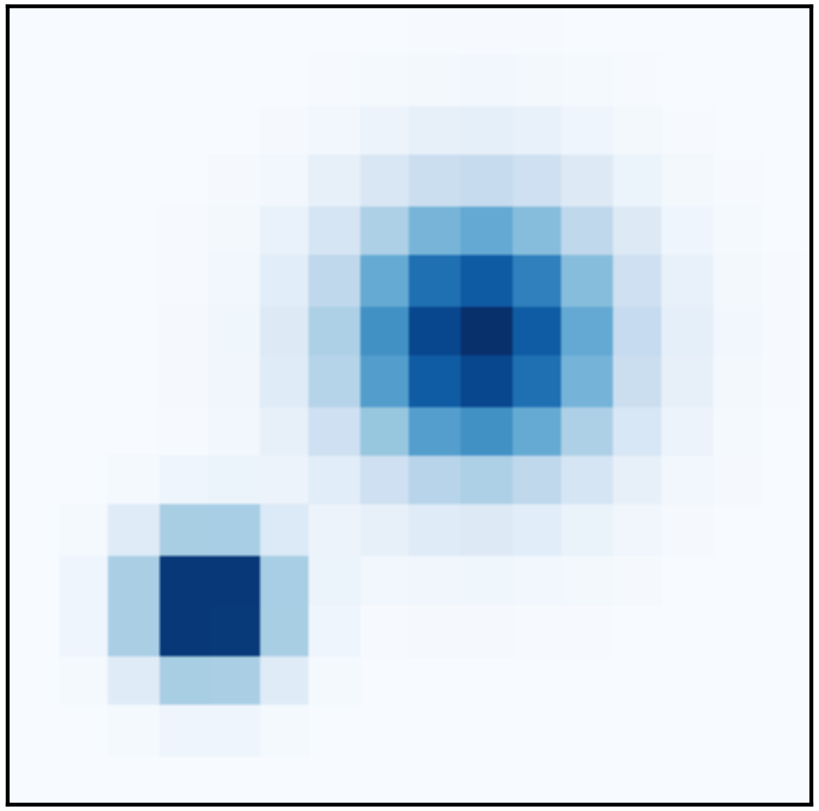
Discrete measure: $\mu = \sum_{i=1}^{n} \mu_i \delta_{\mathbf{x}_i}, \quad \mathbf{x}_i \in \Omega, \quad \sum_{i=1}^{n} \mu_i = 1$

Lagrangian (point clouds)



- Constant weight: $\mu_i = \frac{1}{n}$
- Quotient space: Ω^n, Σ_n

Eulerian (histograms)



- Fixed positions $\mathbf{x}_i$ e.g. grid
- Convex polytope Σ_n (simplex): $\{(\mu_i)_i \geq 0; \sum_i \mu_i = 1\}$

Wasserstein space

The space of probability distribution equipped with the Wasserstein metric ($\mathcal{P}_p(X)$, $W_2^2(X)$) defines a geodesic space with a Riemannian structure [Santambrogio, 2014].

- Geodesics are shortest curves on $\mathcal{P}_p(X)$ that link two distributions

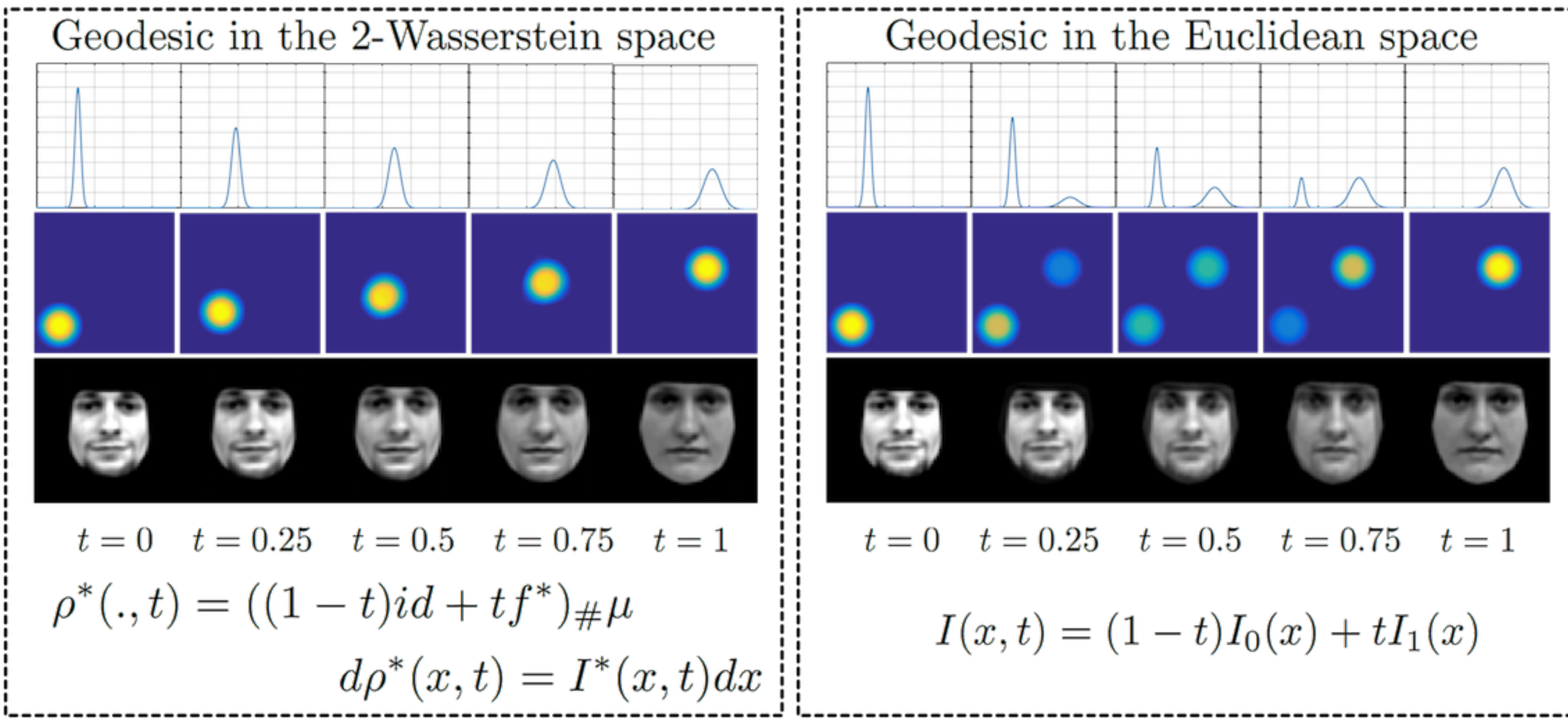
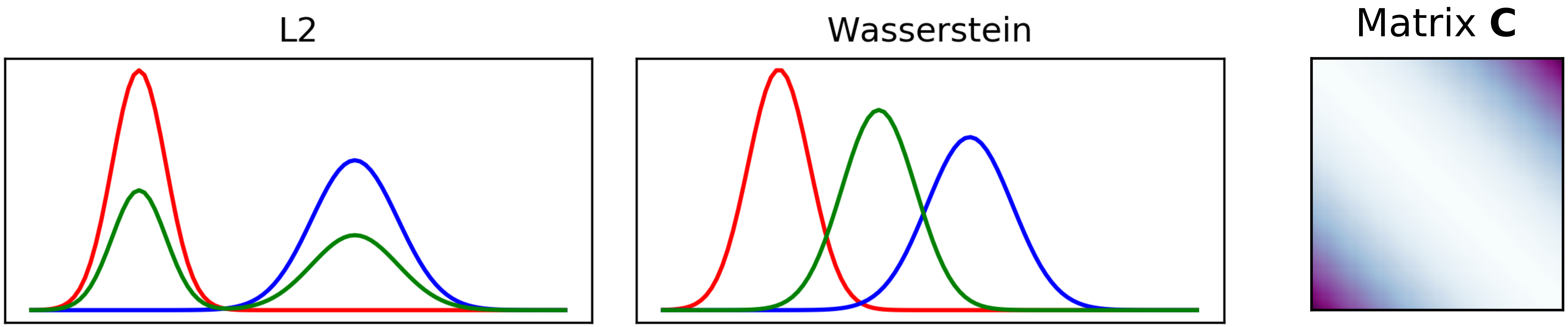


Illustration by S. Kolhouri

Wasserstein barycenter



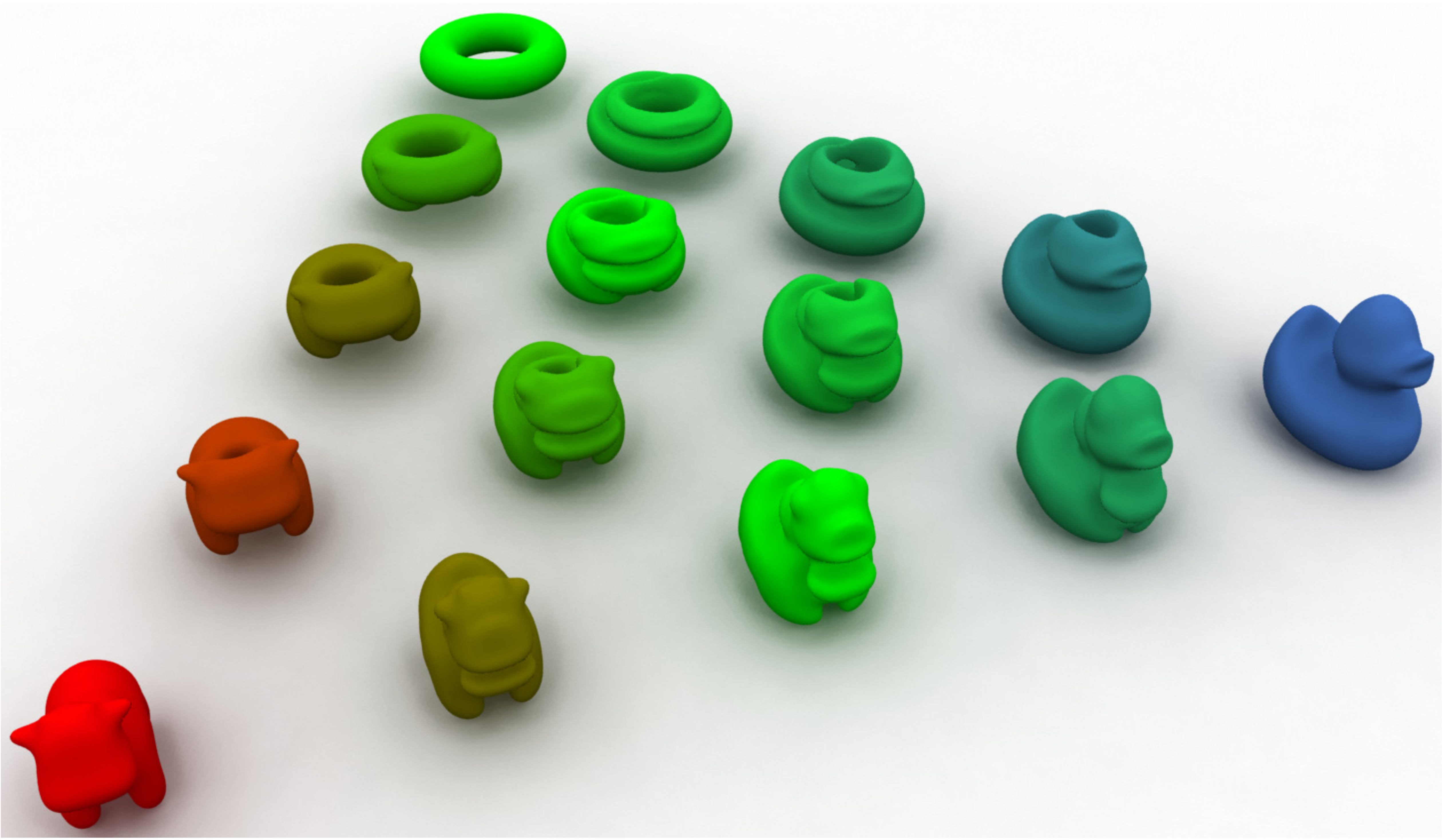
Barycenters [Agueh and Carlier, 2011]

$$\bar{\mu} = \arg\min_{\mu} \quad \sum_{i}^{n} \lambda_i W_p^p(\mu^i, \mu)$$

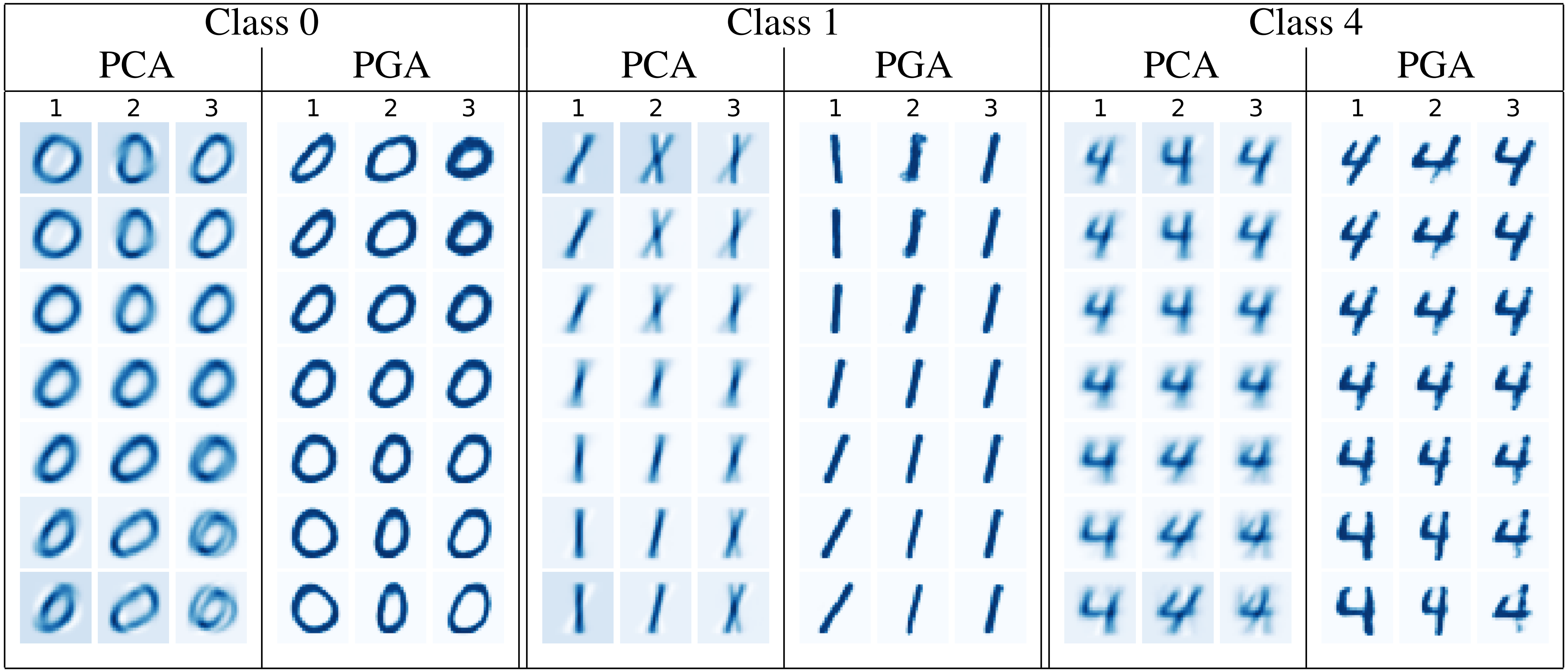
- $\lambda_i > 0$ and $\sum_i^n \lambda_i = 1$.
- Uniform barycenter has $\lambda_i = \frac{1}{n}, \forall i$.
- Interpolation with n=2 and $\boldsymbol{\lambda} = [1 - t, t]$ with $0 \leq t \leq 1$ [McCann, 1997].
- Regularized barycenters using Bregman projections [Benamou et al., 2015].
- The cost and regularization impacts the interpolation trajectory.

3D Wasserstein barycenter

Shape interpolation [Solomon et al., 2015]



Principal Geodesics Analysis



Geodesic PCA in the Wasserstein space [Bigot et al., 2017]

- Generalization of Principal Component Analysis to the Wassertsein manifold.
- Regularized OT [Seguy and Cuturi, 2015].
- Approximation using Wasserstein embedding [Courty et al., 2017].
- Also note recent Wasserstein Dictionary Learning approaches [Schmitz et al., 2017].

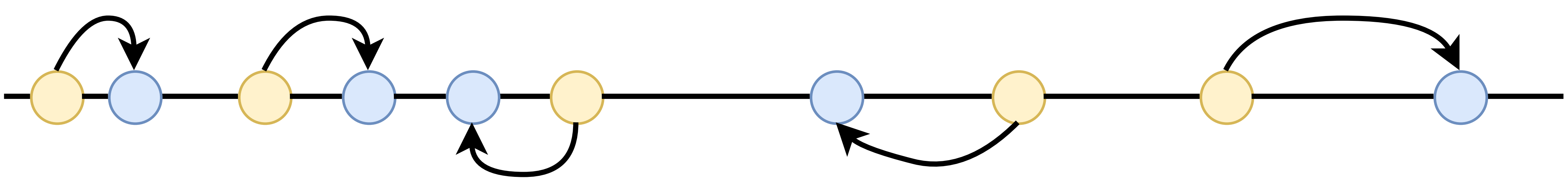
Computational aspects

Special case: 1D distribution

We consider the case where $c(x, y)$ is a strictly convex and increasing function of $|x - y|$.

- if $x_1 < x_2$ and $y_1 < y_2$, it is easy to check that $c(x_1, y_1) + c(x_2, y_2) < c(x_1, y_2) + c(x_2, y_1)$
- As such, any optimal transport plan respects the ordering of the elements, and the solution is given by the monotone rearrangement of μ_1 onto μ_2

This gives very simple algorithm to compute the transport in $O(N \log N)$, by sorting both x_i and y_i and summing the absolute values of differences.



Special case: 1D distribution

Consider the cumulative distribution functions F_μ associated to the μ distribution.

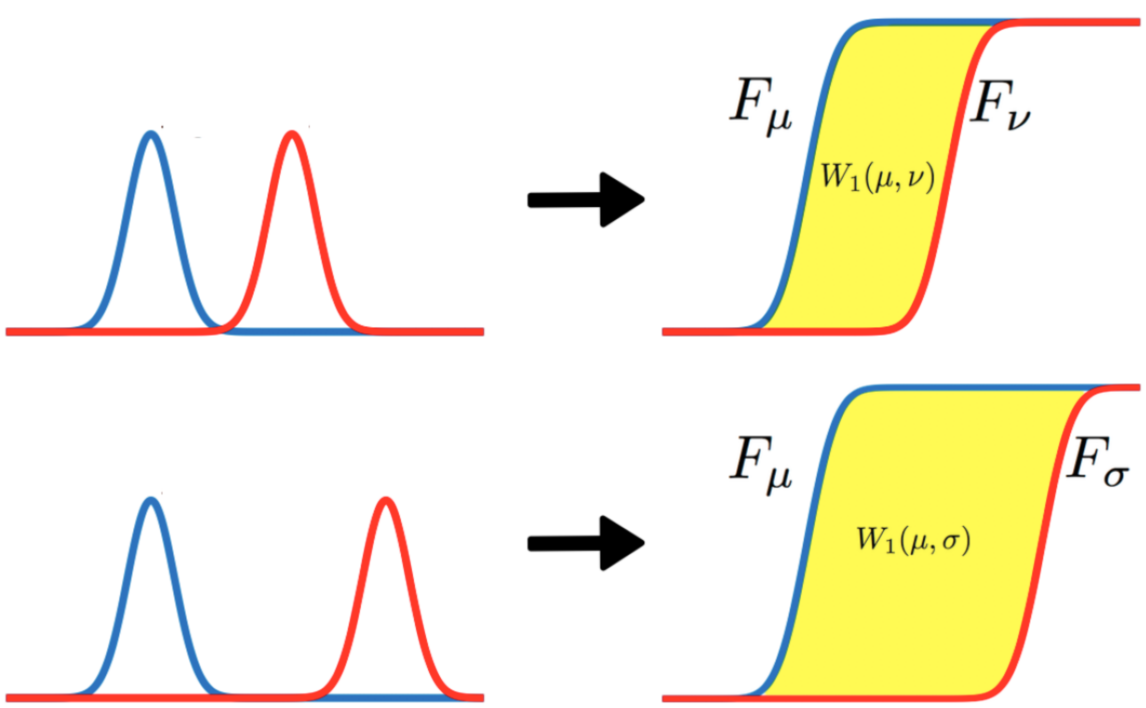
- It is defined such that $F_\mu(t) = \mu(-\infty, t]$.

We will note $F_\mu^{-1}(q),\ q \in [0,1]$ the corresponding generalized inverse distribution (or quantile function)

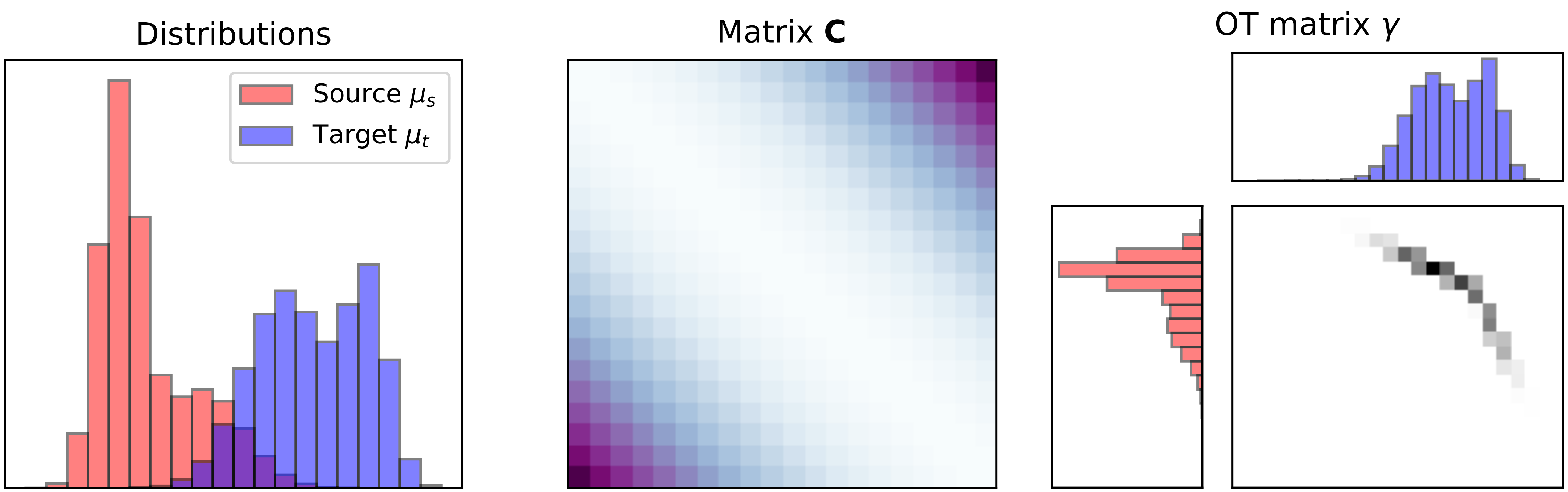
- defined as $F_\mu^{-1}(q) = \inf\{x \in \mathbb{R} : F_\mu(x) \geq q\}$.

Then,

$$W_1(\mu_s, \mu_t) = \int_0^1 c(F_{\mu_s}^{-1}(q), F_{\mu_t}^{-1}(q))dq$$



Optimal transport with discrete distributions



OT Linear Program

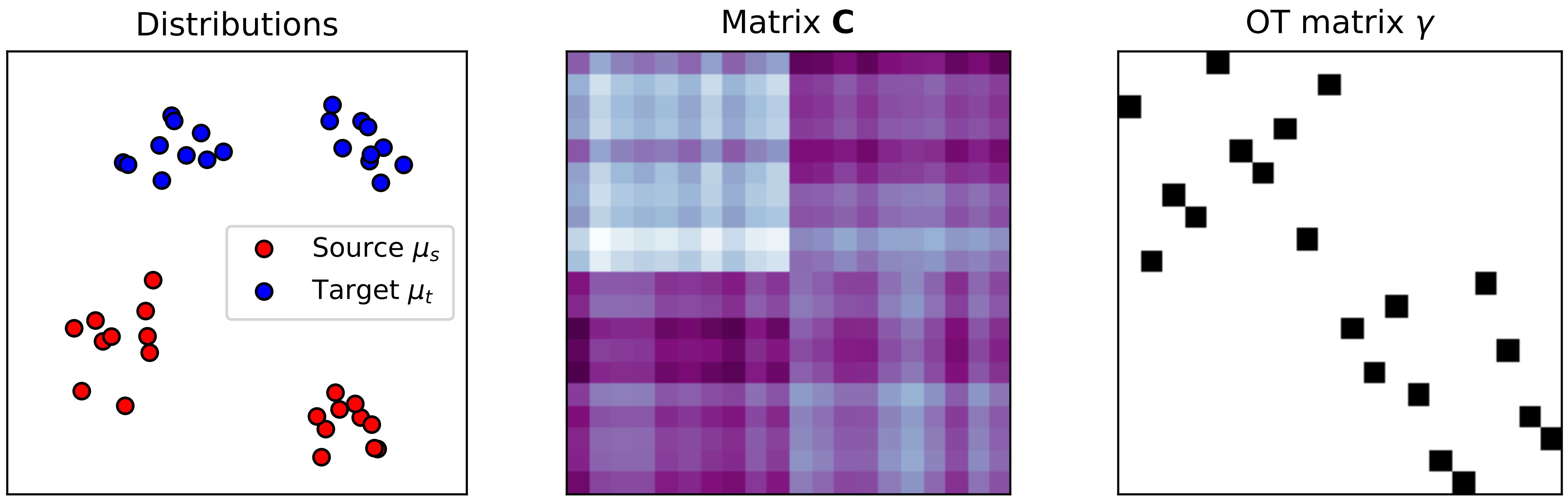
$$\gamma_0 = \underset{\gamma \in \mathcal{P}}{\operatorname{argmin}} \quad \left\{ \langle \gamma, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

where $\mathbf{C}$ is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

$$\mathcal{P} = \left\{ \gamma \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} | \, \gamma \mathbf{1}_{\mathbf{n_t}} = \mu_{\mathbf{s}}, \gamma^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mu_{\mathbf{t}} \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

Optimal transport with discrete distributions



OT Linear Program

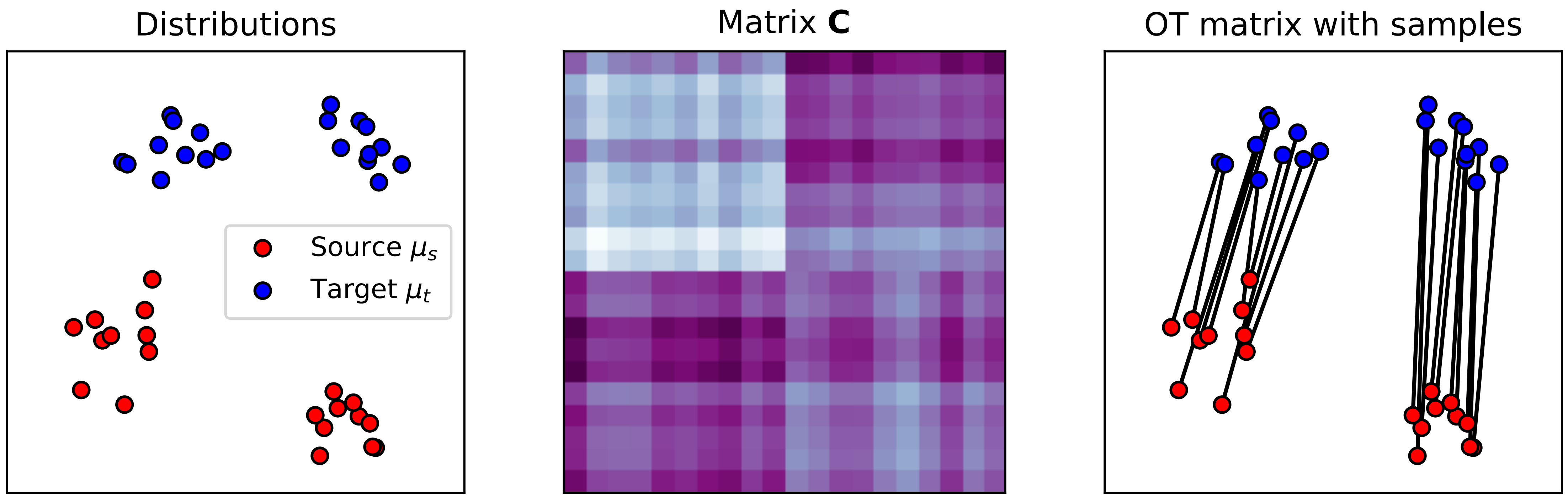
$$\gamma_0 = \underset{\gamma \in \mathcal{P}}{\operatorname{argmin}} \quad \left\{ \langle \gamma, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

where $\mathbf{C}$ is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

$$\mathcal{P} = \left\{ \gamma \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} | \, \gamma \mathbf{1}_{\mathbf{n_t}} = \mu_\mathbf{s}, \gamma^\mathbf{T} \mathbf{1}_{\mathbf{n_s}} = \mu_\mathbf{t} \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

Optimal transport with discrete distributions



OT Linear Program

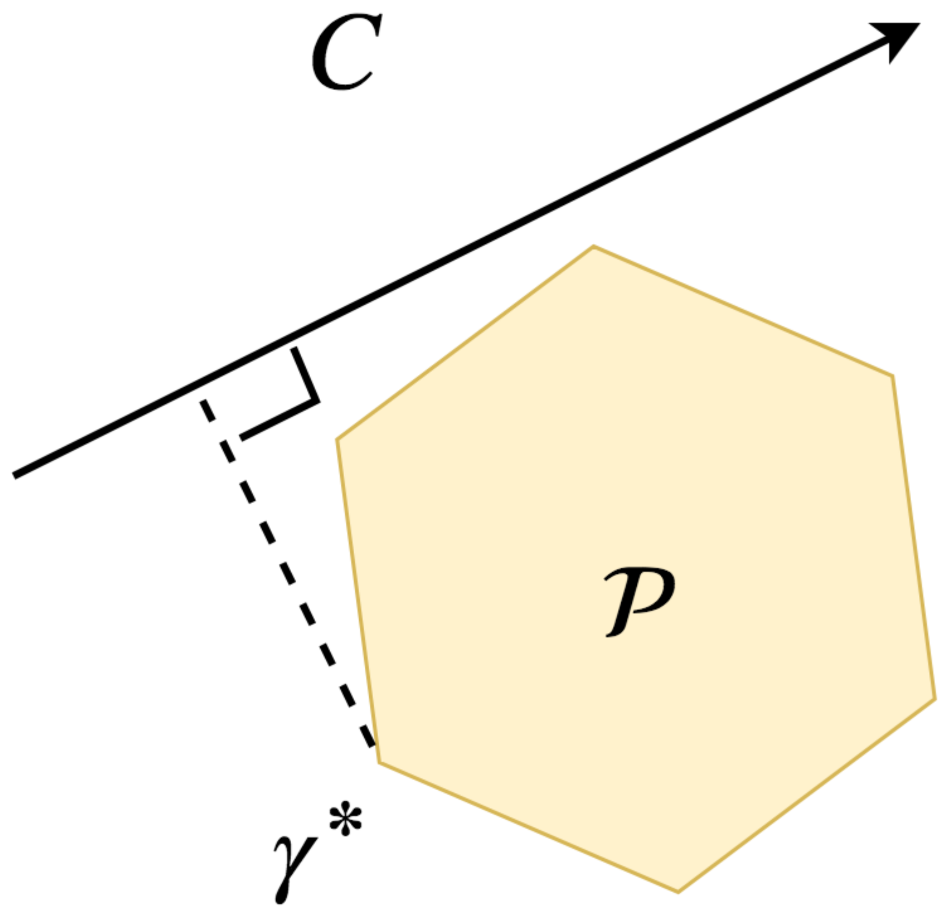
$$\gamma_0 = \underset{\gamma \in \mathcal{P}}{\text{argmin}} \quad \left\{ \langle \gamma, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

where $\mathbf{C}$ is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

$$\mathcal{P} = \left\{ \gamma \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} | \, \gamma \mathbf{1}_{\mathbf{n_t}} = \mu_\mathbf{s}, \gamma^\mathbf{T} \mathbf{1}_{\mathbf{n_s}} = \mu_\mathbf{t} \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

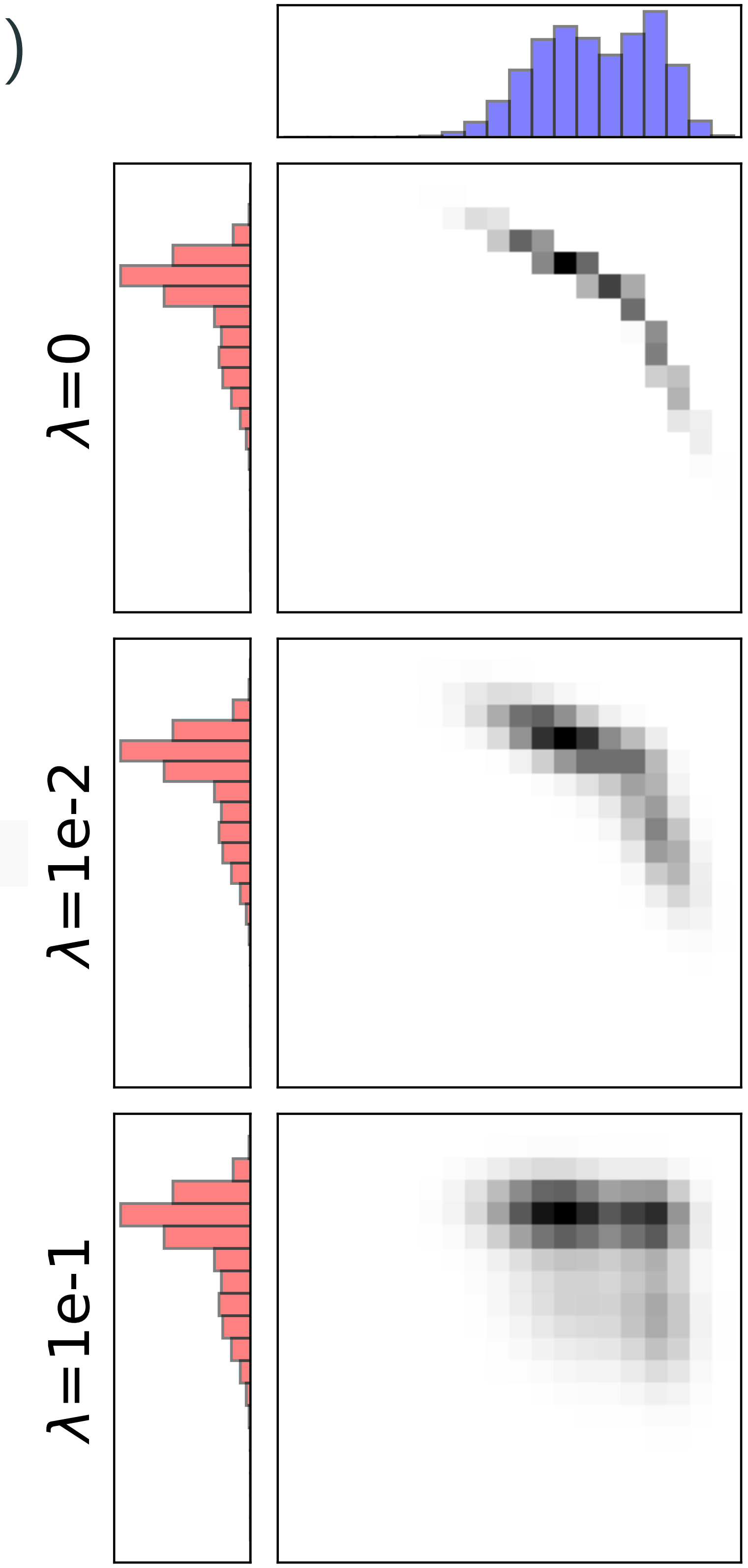
Optimal transport with discrete distributions



- $\mathcal{P}$ is the Birkhoff polytope
- No unique solution in some cases, numerical instabilities
- Not differentiable !

Regularized optimal transport

$$\gamma_0^\lambda = \underset{\gamma \in \mathcal{P}}{\operatorname{argmin}} \quad \langle \gamma, \mathbf{C} \rangle_F + \lambda \Omega(\gamma), \tag{4}$$



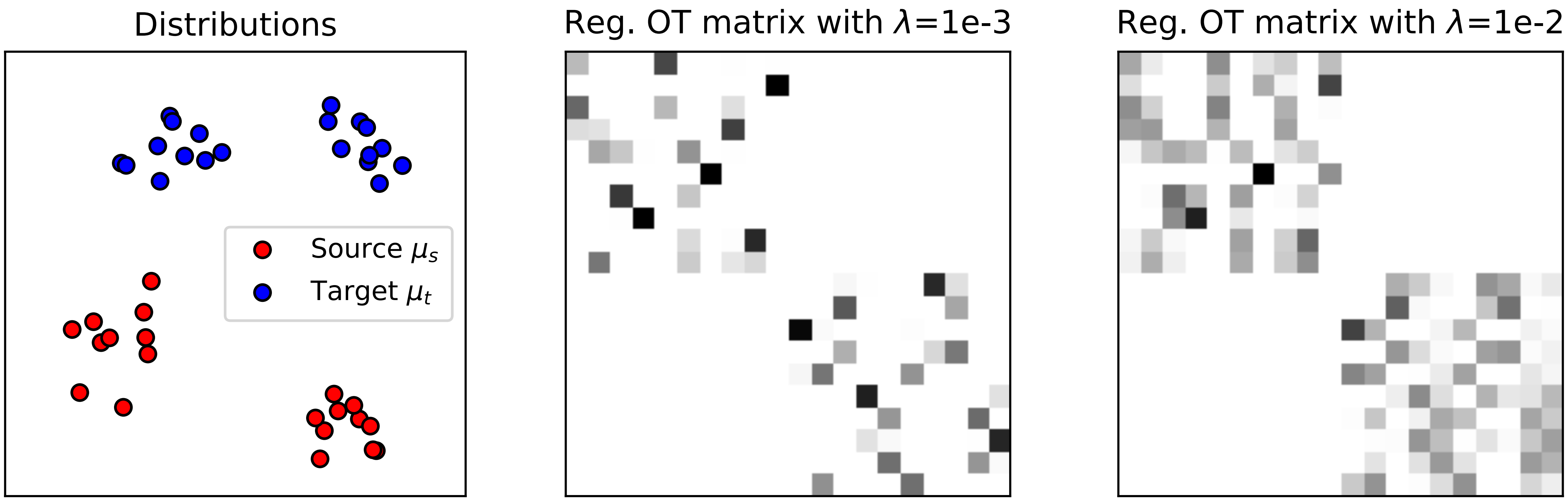
Regularization term $\Omega(\gamma)$

- Entropic regularization [Cuturi, 2013].
- Group Lasso [Courty et al., 2016].
- KL, Itakura Saito, β-divergences, [Dessein et al., 2016].

Why regularize?

- Smooth the "distance" estimation:
  $$W_\lambda(\mu_s, \mu_t) = \langle \gamma_0^\lambda, \mathbf{C} \rangle_F$$
- Encode prior knowledge on the data.
- Better posed problem (convex, stability).
- Fast algorithms to solve the OT problem.

Entropic regularized optimal transport

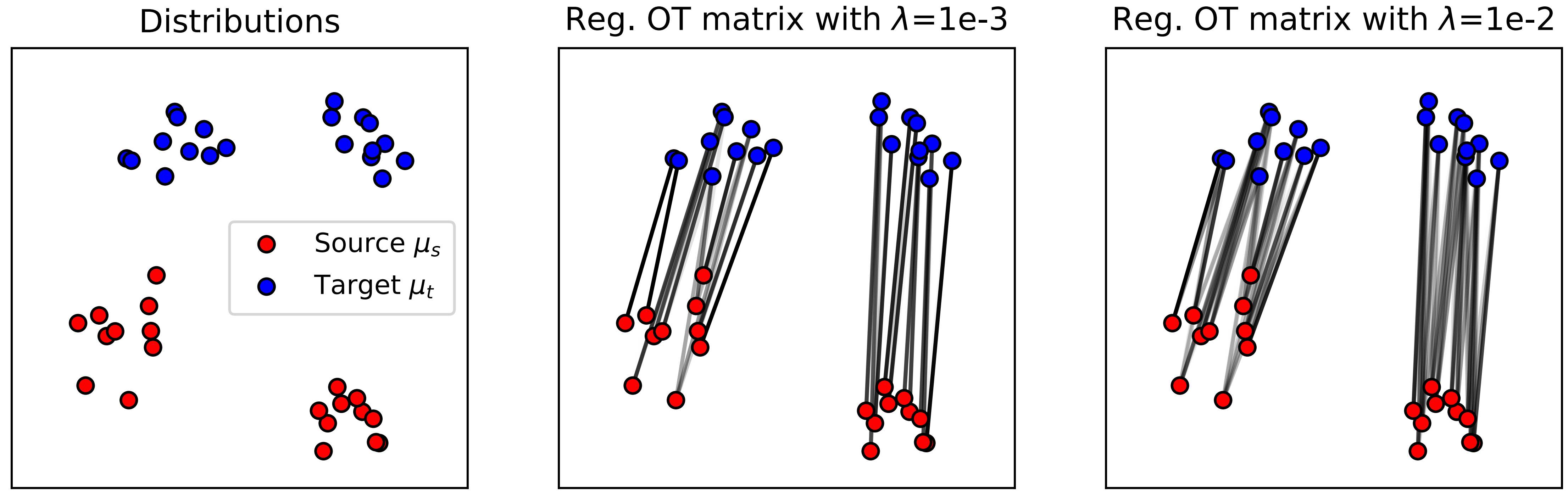


Entropic regularization [Cuturi, 2013]

$$\Omega(\gamma) = \sum_{i,j} \gamma(i,j)(\log \gamma(i,j) - 1)$$

- Regularization with the negative entropy of γ.

Entropic regularized optimal transport



Entropic regularization [Cuturi, 2013]

$$\Omega(\boldsymbol{\gamma}) = \sum_{i,j} \boldsymbol{\gamma}(i,j)(\log \boldsymbol{\gamma}(i,j) - 1)$$

- Regularization with the negative entropy of γ.

Resolving the entropy regularized problem

Entropy-regularized transport

The solution of entropy regularized optimal transport problem is of the form
$$\gamma_0^\lambda = \text{diag}(\mathbf{u}) \exp(-\mathbf{C}/\lambda)\text{diag}(\mathbf{v})$$

Why ? Consider the Lagrangian of the optimization problem:

$$\mathcal{L}(\gamma, \alpha, \beta) = \sum_{ij} \gamma_{ij} \mathbf{C}_{ij} + \lambda \gamma_{ij} (\log \gamma_{ij} - 1) + \alpha^{\mathbf{T}}(\gamma \mathbf{1}_{n_t} - \mu_s) + \beta^{\mathbf{T}}(\gamma^T \mathbf{1}_{n_s} - \mu_t)$$

$$\begin{aligned} \partial \mathcal{L}(\gamma, \alpha, \beta)/\partial \gamma_{ij} &= \mathbf{C}_{ij} + \lambda \log \gamma_{ij} + \alpha_i + \beta_j \\ \partial \mathcal{L}(\gamma, \alpha, \beta)/\partial \gamma_{ij} = 0 \quad &\Longrightarrow \quad \gamma_{ij} = \exp(\frac{\alpha_i}{\lambda}) \exp(-\frac{\mathbf{C}_{ij}}{\lambda}) \exp(\frac{\beta_j}{\lambda}) \end{aligned}$$

Not now !

- Through the **Sinkhorn theorem** $\text{diag}(\mathbf{u})$ and $\text{diag}(\mathbf{v})$ exist and are unique.
- Can be solved by the **Sinkhorn-Knopp** algorithm (implementation in parallel, GPU).

Dual formulation of optimal transport

- Yet, solving for γ is impractical to intractable when dealing with high-dimensional distributions
- especially if one is interested in computing the gradients of the Wasserstein distance
- Other solving strategies should be taken into consideration
- Recalling that any LP problem can be turnt into its dual form:

$$\begin{array}{rrcl|rrcl} & \multicolumn{3}{l|}{\textbf{primal form:}} & & \multicolumn{3}{l}{\textbf{dual form:}} \\ \text{minimize} & z & = & \mathbf{c}^T\mathbf{x}, & \text{maximize} & \tilde{z} & = & \mathbf{b}^T\mathbf{y}, \\ \text{so that} & \mathbf{A}\mathbf{x} & = & \mathbf{b} & \text{so that} & \mathbf{A}^T\mathbf{y} & \leq & \mathbf{c} \\ \text{and} & \mathbf{x} & \geq & \mathbf{0} & & & & \end{array}$$

- Weak duality: $\tilde{z}$ is a lower bound of z, Strong duality $\tilde{z} = z$
- Strong duality is usually achieved via Farkas Theorem

Duality: general case with continuous distributions

We now introduce two functions scalar functions ϕ and ψ (also known as Kantorovich potentials) that will act as our dual variables. Then, we consider the optimal problem is equivalent (by the Rockafellar-Fenchel theorem) to:

$$\max_{\phi,\psi} \left\{ \int \phi d\mu_s + \int \psi d\mu_t \;\middle|\; \phi(x) + \psi(y) \leq c(x,y) \right\} \tag{6}$$

Note that the marginal constraint has been turned into an equality constraint on ϕ and ψ

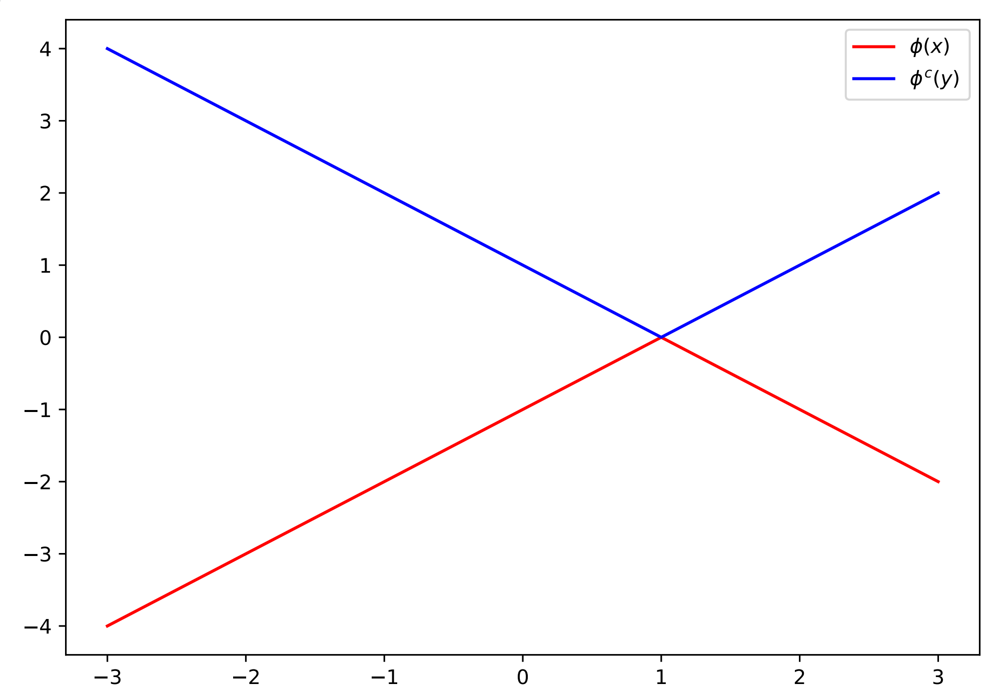
Introducing the *c-transform* (or *c-conjugate*) H^c which is in spirit close to a Legendre transform:

$$\phi^c \stackrel{\text{def}}{=} H^c(\phi) = \inf_x c(x,y) - \phi(x) \tag{7}$$

then the following problem is equivalent:

$$\max_{\phi} \left\{ \int \phi d\mu_s + \int \phi^c d\mu_t \;\middle|\; \phi(x) + \phi^c(y) \leq c(x,y) \right\} \tag{8}$$

Case $c(x, y) = |x - y|$ (a.k.a W_1^1)



Whenever $c(x, y) = |x - y|$, then:

- existence of a solution but not unique
- For any $\phi \in \text{Lip}^1$ (set of 1-Lipschitz functions), we have $\phi^c(x) = -\phi(x)$

The optimal transport problem then amounts to find $\phi \in \text{Lip}^1$ as

$$\sup_{\phi \in \text{Lip}^1} \int \phi d(\mu_s - \mu_t) = \sup_{\phi \in \text{Lip}^1} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mu_s} [\phi(x)] - \mathop{\mathbb{E}}_{\mathbf{y} \sim \mu_t} [\phi(y)] \tag{9}$$

- also known as **Kantorovich-Rubinstein duality**
- ϕ can be learnt as a neural network constrained to the set Lip^1, see next section on GAN

Dual: empirical version

In the case when we have access to discrete distributions, μ_s (resp. μ_t) is characterized by a set of locations $\mathbf{X^s}$ and masses $\mathbf{a} \in \mathbb{R}^{n^s}$ (resp. $\mathbf{X^t}$ and $\mathbf{b} \in \mathbb{R}^{n^t}$)

Discrete dual version of OT

$$W(\mu_s, \mu_t) = \max_{\alpha \in \mathbb{R}^{n^s}, \beta \in \mathbb{R}^{n^t}, \alpha_i + \beta_j \leq c(\mathbf{X_i^s}, \mathbf{X_j^t})} \alpha^T \mathbf{a} + \beta^T \mathbf{b} \tag{11}$$

i.e. find a scalar values per sample

Regularized case

Adding regularization to the original problem turns the dual computation to an unconstrained problem !

In the case of entropy regularization, *i.e.*

$W_\lambda(\mu_s, \mu_t) = \min_{\gamma \in \mathcal{P}} \quad \langle \boldsymbol{\gamma}, \mathbf{C} \rangle_F + \lambda \Omega(\boldsymbol{\gamma})$ with $\Omega(\boldsymbol{\gamma}) = \sum_{i,j} \boldsymbol{\gamma}(i,j) \log \boldsymbol{\gamma}(i,j)$,

the dual now reads (in a discrete settings, measures are collections of Diracs):

$$\max_{\alpha, \beta} \alpha^T \mu_s + \beta^T \mu_t - \frac{1}{\lambda} \exp(\frac{\alpha}{\lambda})^T \mathbf{K} \exp(\frac{\beta}{\lambda}) \tag{12}$$

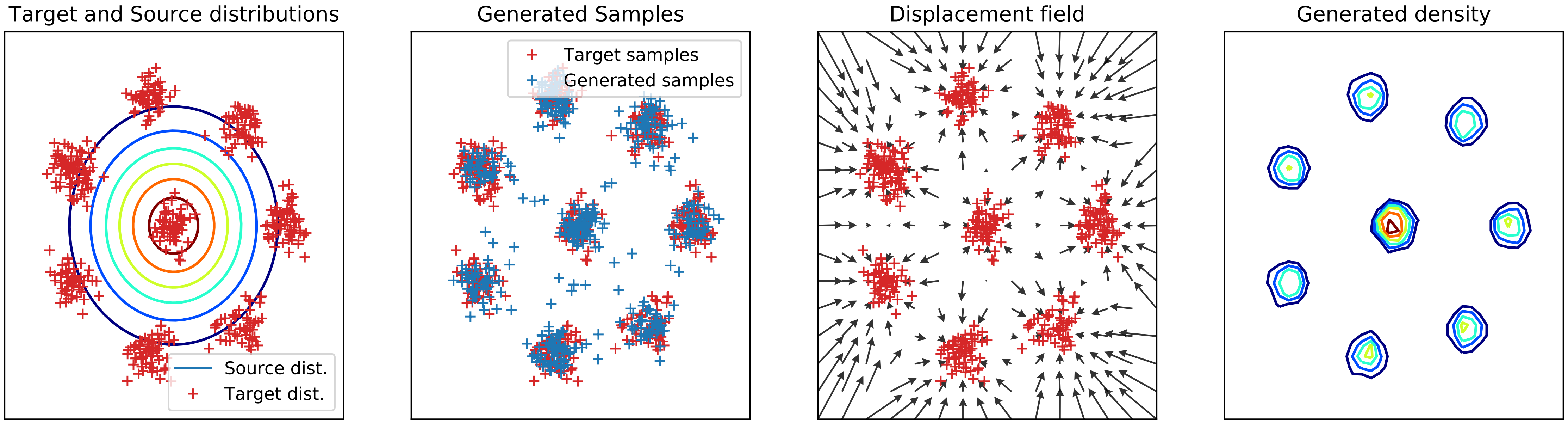
with $\mathbf{K} = \exp(-\frac{\mathbf{C}}{\lambda})$.

Remark: The Sinkhorn algorithm is a gradient ascent on the dual variables !

Regularized case

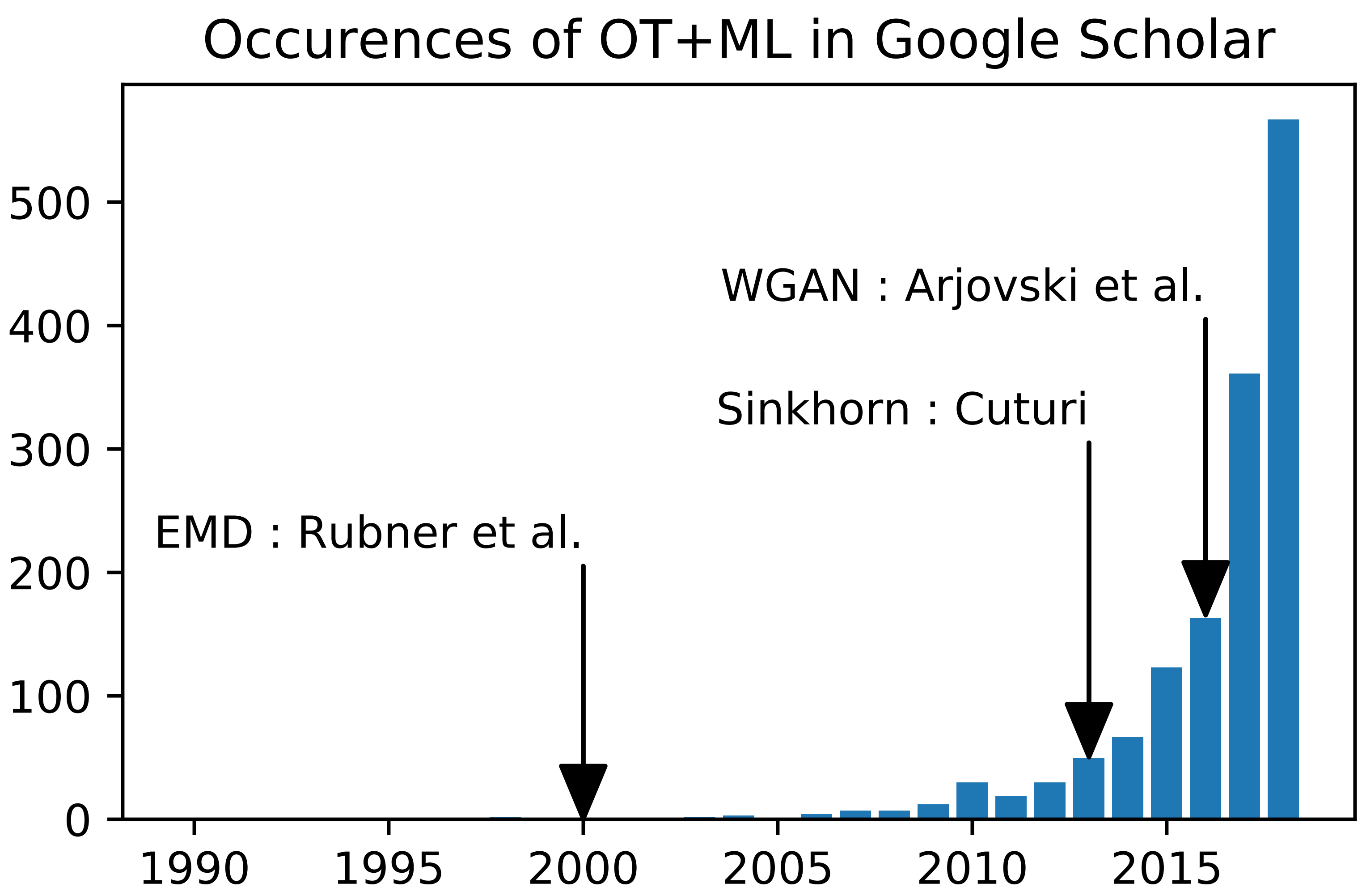
With this unconstrained problem, incremental gradients techniques (SGD, SAG) can be used to solve the problem !

- [Genevay et al., 2016] used the semi-dual formulation (one variable is removed by replacing it with its c-transform) int the first stochastic version of Optimal Transport problem
- [Seguy et al., 2017] used the full dual version with entropic and L2 regularizations, together with neural networks to parameterize the problem.



1. What is optimal transport ?
2. How can it be used in data science ?

Optimal transport for machine learning

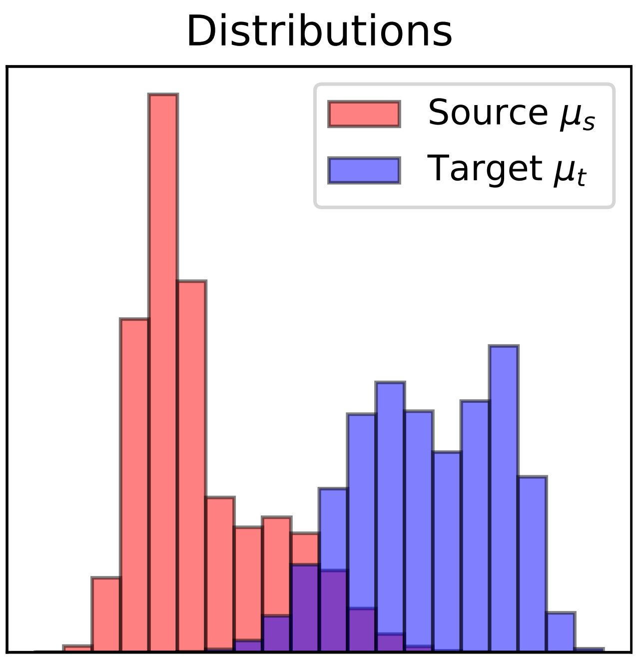


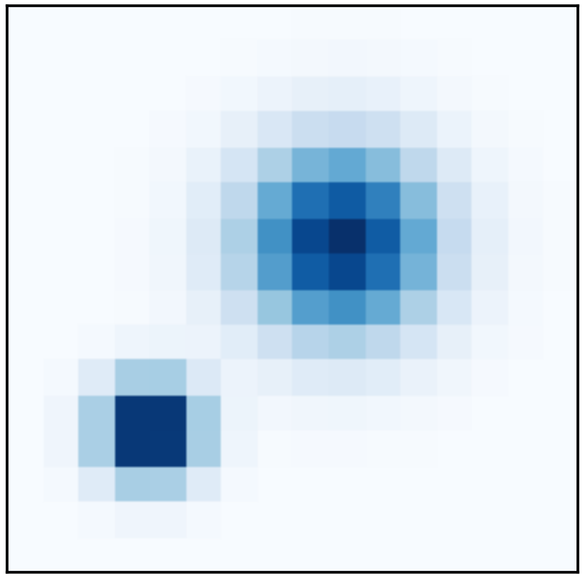
Short history of OT for ML

- Recently introduced to ML (well known in image processing since 2000s).
- Computationnal OT allow numerous applications (regularization).
- Deep learning boost (numerical optimization and GAN).

Learning from histograms

Learning from histograms

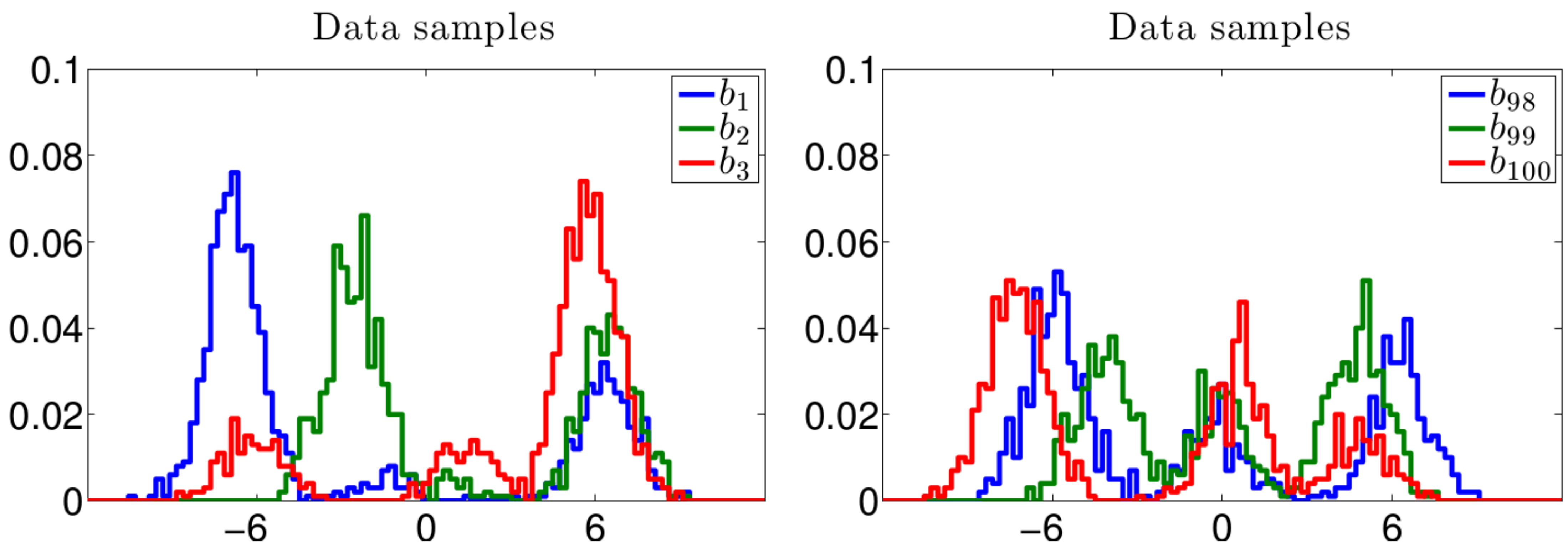




Data as histograms

- Fixed bin positions $\mathbf{x}_i$ e.g. grid, simplex $\Delta = \{(\mu_i)_i \geq 0; \sum_i \mu_i = 1\}$
- A lot of datasets comes under the form of histograms.
- Images are photo counts (black and white), text as word counts.
- Natural divergence is Kullback–Leibler.
- Not all data can be seen as histograms (positivity+constant mass)!

Dictionary learning on histograms

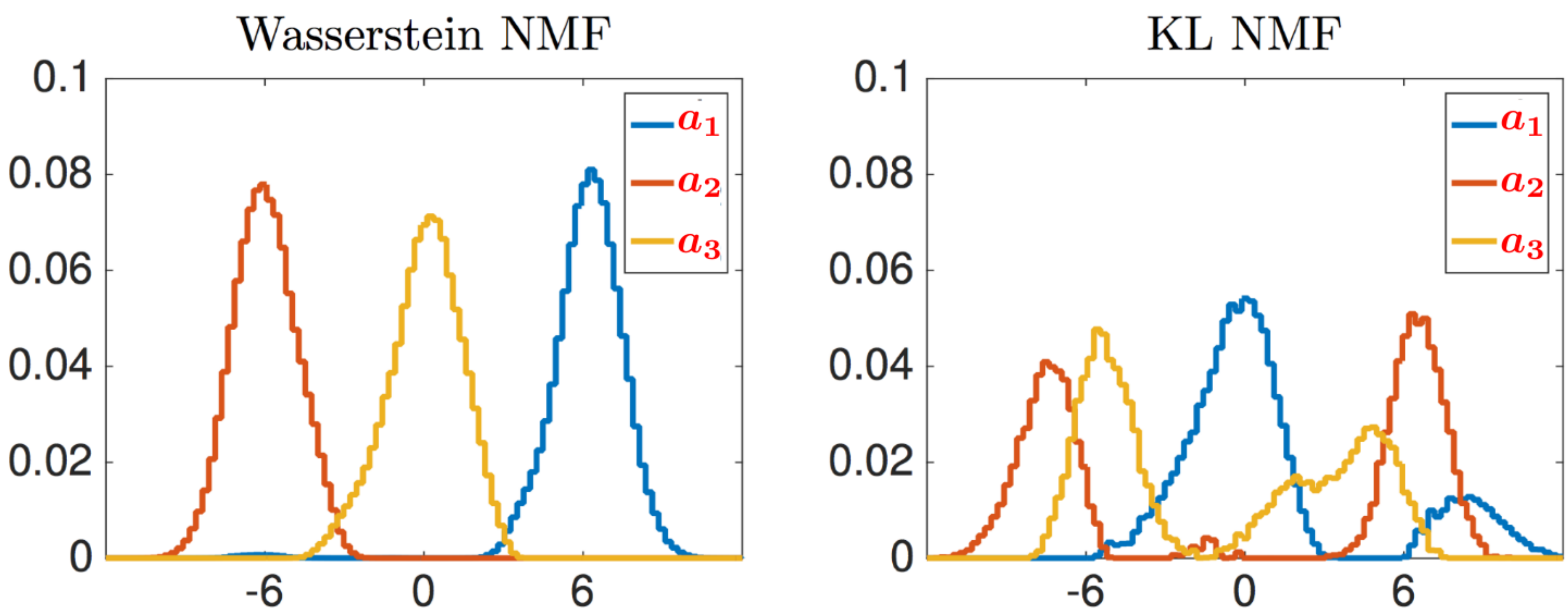


DL with Wasserstein distance [Sandler and Lindenbaum, 2011]

$$\min_{\mathbf{D},\mathbf{H}} \quad \sum_i W_{\mathbf{C}}(\mathbf{v}_i, \mathbf{D}\mathbf{h}_i)$$

- NMF: columns of $\mathbf{D}$ and $\mathbf{H}$ are on the simplex.
- Metric $\mathbf{C}$ can encode spatial relations btewwen the bins of the histograms.
- Ground metric learning [Zen et al., 2014].
- Fast DL with regularized OT [Rolet et al., 2016].

Dictionary learning on histograms



DL with Wasserstein distance [Sandler and Lindenbaum, 2011]

$$\min_{\mathbf{D},\mathbf{H}} \quad \sum_i W_{\mathbf{C}}(\mathbf{v}_i, \mathbf{D}\mathbf{h}_i)$$

- NMF: columns of $\mathbf{D}$ and $\mathbf{H}$ are on the simplex.
- Metric $\mathbf{C}$ can encode spatial relations btewwen the bins of the histograms.
- Ground metric learning [Zen et al., 2014].
- Fast DL with regularized OT [Rolet et al., 2016].

Multi-label learning with Wasserstein Loss

Siberian husky

Eskimo dog

Flickr : street, parade, dragon
Prediction : people, protest, parade

Flickr : water, boat, ref ection, sun-shine
Prediction : water, river, lake, summer;

Learning with a Wasserstein Loss [Frogner et al., 2015]

$$\min_{f} \quad \sum_{k=1}^{N} W_1^1(f(\mathbf{x}_i), \mathbf{l}_i)$$

- Empirical loss minimization with Wasserstein loss.
- Multi-label prediction (labels $\mathbf{l}$ seen as histograms, f output softmax).
- Cost between labels can encode semantic similarity between classes.
- Good performances in image tagging.

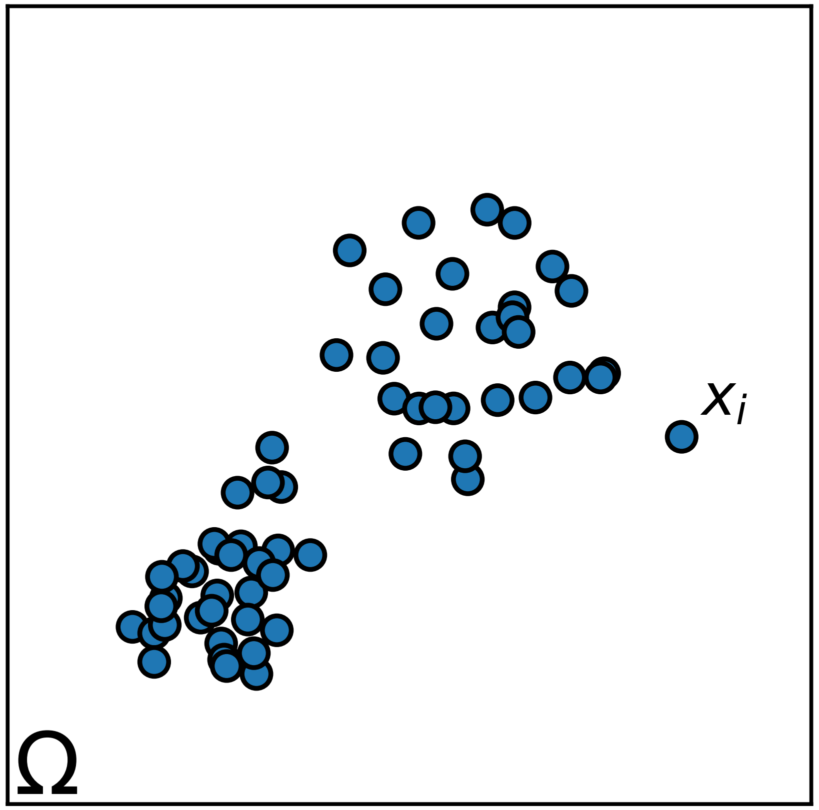
Learning from distributions

Empirical distributions A.K.A datasets

$$\mu = \sum_{i=1}^{n} \mu_i \delta_{\mathbf{x}_i}, \quad \mathbf{x}_i \in \Omega, \quad \sum_{i=1}^{n} \mu_i = 1$$

Empirical distribution

- Two realizations never overlap.
- Training base of all machine learning approaches.
- How to measure discrepancy?
- Maximum Mean Discrepancy (ℓ_2 after convolution).
- Wasserstein distance.



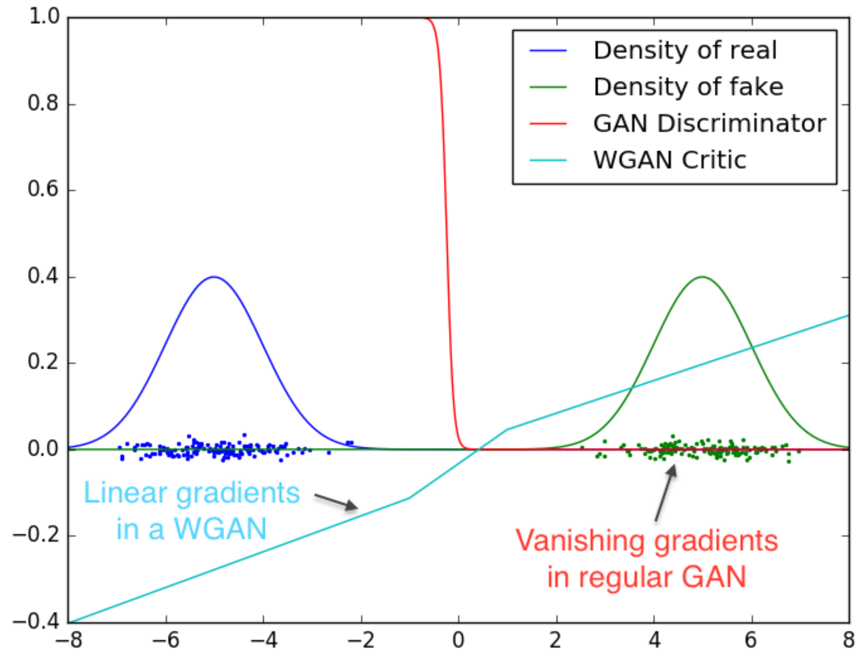
Generative Adversarial Networks (GAN)

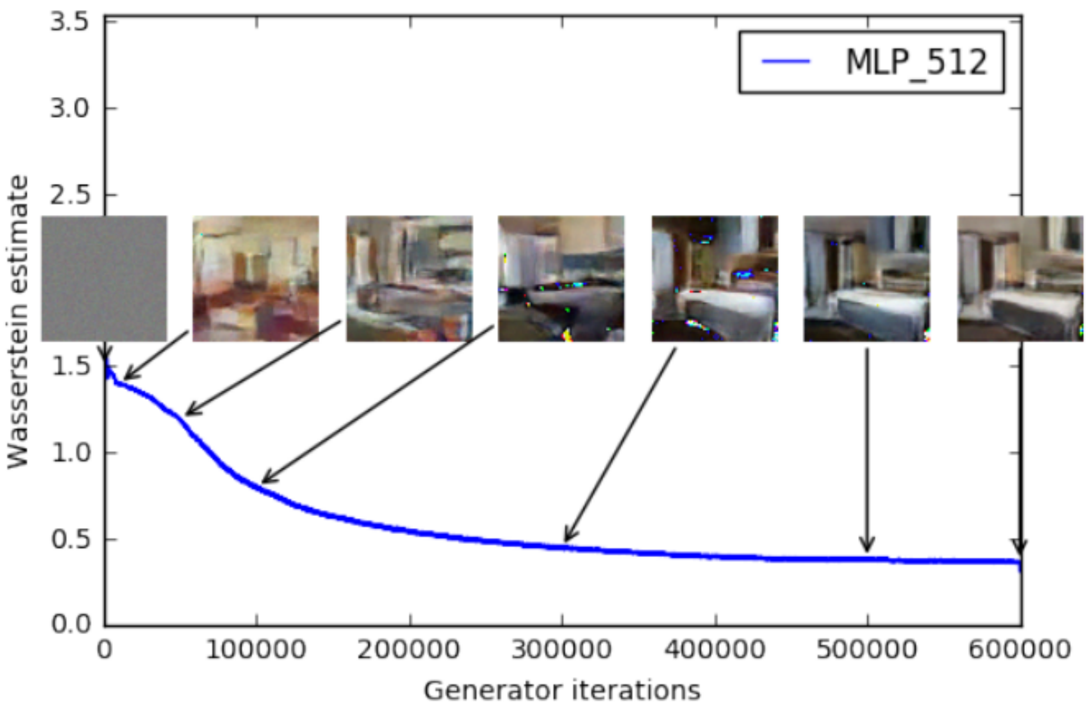
Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]

$$\min_G \max_D \quad E_{\mathbf{x}\sim\mu_d}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})}[\log(1 - D(G(\mathbf{z})))]$$

- Learn a generative model G that outputs realistic samples from data μ_d.
- Learn a classifier D to discriminate between the generated and true samples.
- Make those models compete (Nash equilibrium [Zhao et al., 2016]).
- Generator space has semantic meaning [Radford et al., 2015].
- But extremely hard to train (vanishing gradients).

Wasserstein Generative Adversarial Networks (WGAN)





Wasserstein GAN [Arjovsky et al., 2017]

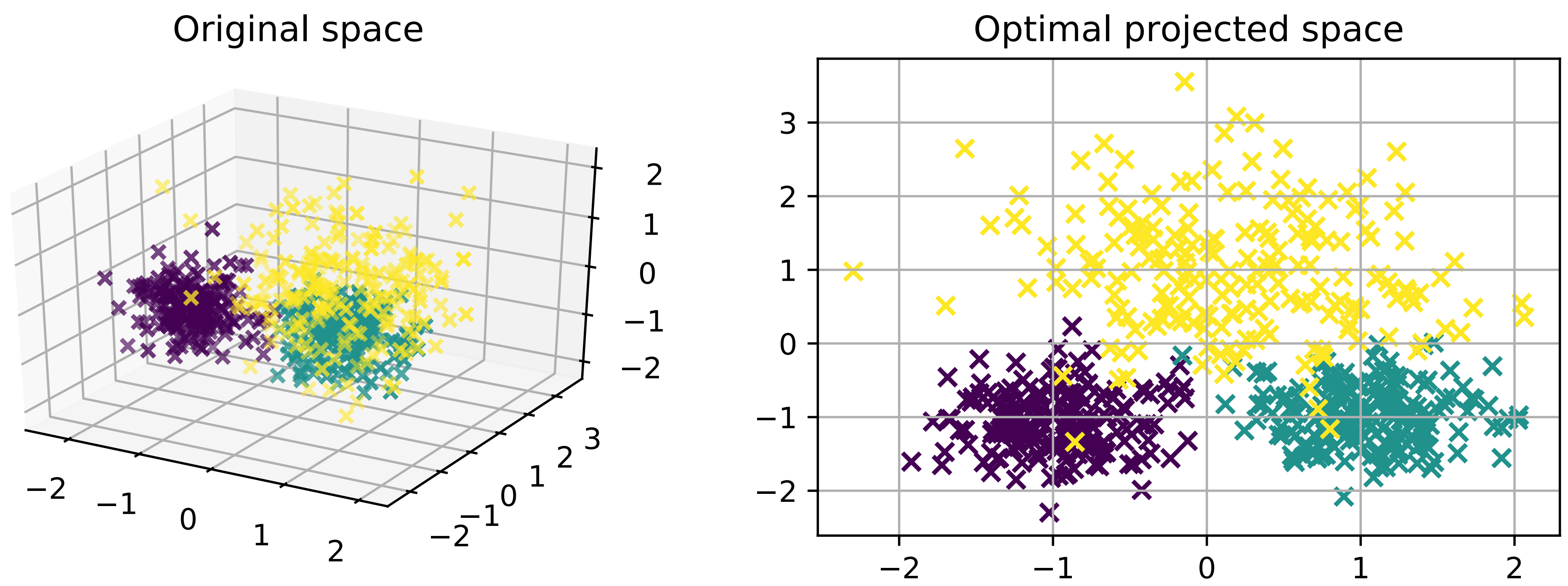
$$\min_{G} \quad W_1^1(G(\mathbf{z}), \mu_d), \quad \text{s.t. } \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{3}$$

- Minimizes the Wasserstein distance between the data and the generated data.
- No vanishing gradients ! Far better convergence in practice.
- Wasserstein in the dual (separable w.r.t. the samples).

$$\min_{G} \sup_{\phi \in \text{Lip}^1} \quad \mathbb{E}_{\mathbf{x}\sim\mu_d}[\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})}[\phi(G(\mathbf{z}))]$$

- ϕ is a neural network that acts as an *actor critic*

Wasserstein Discriminant Analysis (WDA)

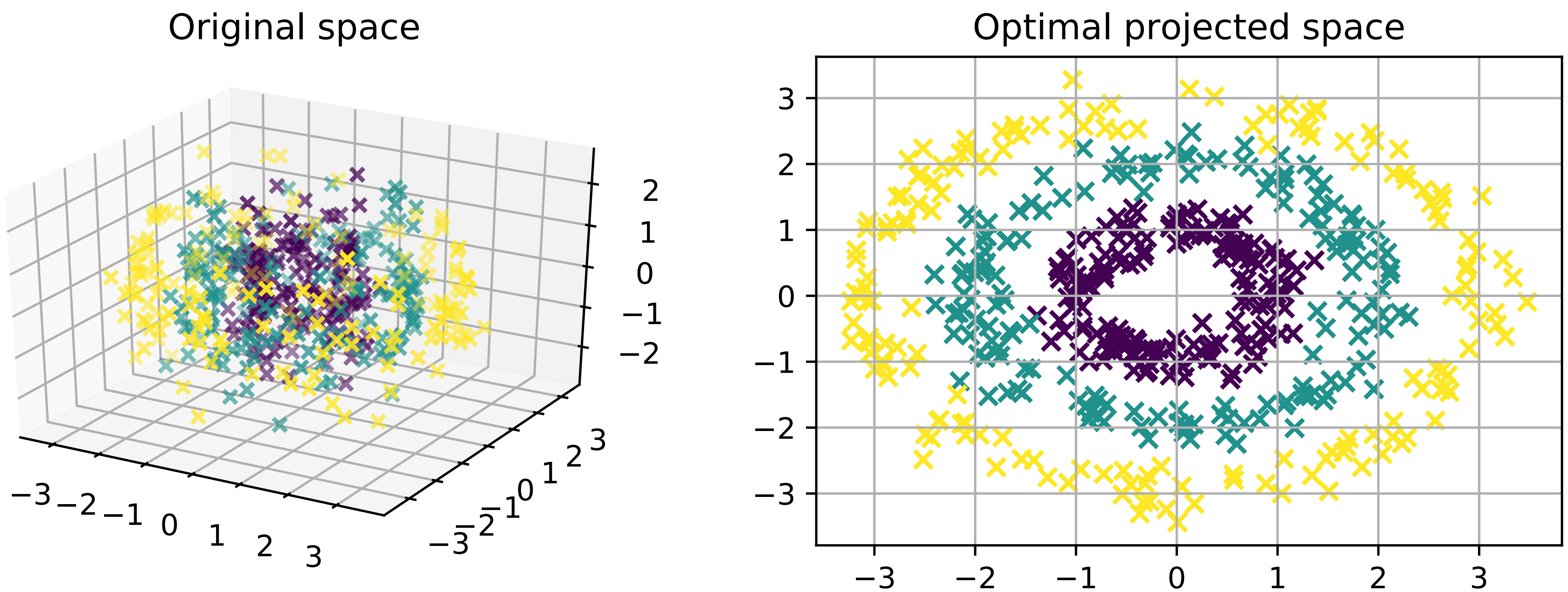


$$\max_{\mathbf{P}\in\mathcal{S}} \quad \frac{\sum_{c,c'>c} W_\lambda(\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^{c'})}{\sum_c W_\lambda(\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^c)} \quad (4)$$

- $\mathbf{X}^c$ are samples from class c.
- $\mathbf{P}$ is an orthogonal projection;

- Converges to Fisher Discriminant when $\lambda \to \infty$.
- Non parametric method that allows nonlinear discrimination.
- Problem solved with gradient ascent in the Stiefel manifold $\mathcal{S}$.
- Gradient computed using automatic differentiation of Sinkhorn algorithm.

Wasserstein Discriminant Analysis (WDA)



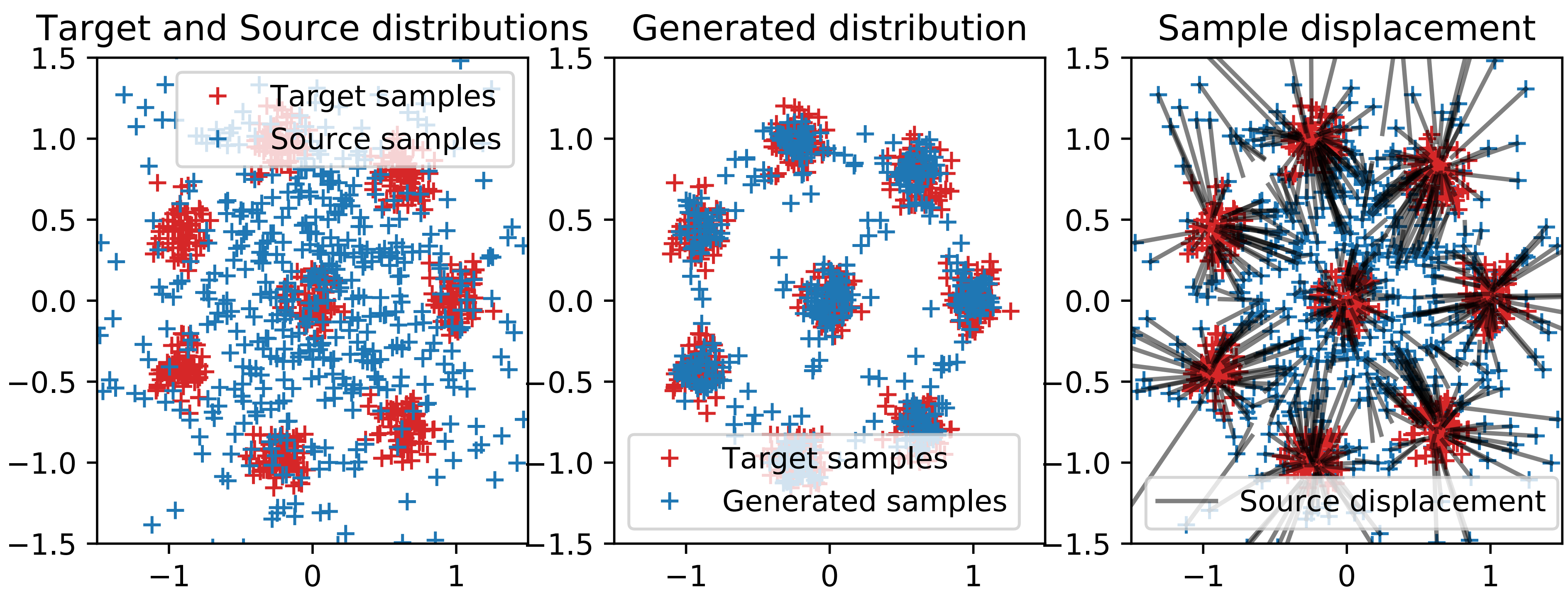
$$\max_{\mathbf{P}\in\mathcal{S}} \quad \frac{\sum_{c,c'>c} W_\lambda(\mathbf{PX}^c, \mathbf{PX}^{c'})}{\sum_c W_\lambda(\mathbf{PX}^c, \mathbf{PX}^c)} \quad (4)$$

- $\mathbf{X}^c$ are samples from class c.
- $\mathbf{P}$ is an orthogonal projection;

- Converges to Fisher Discriminant when $\lambda \to \infty$.
- Non parametric method that allows nonlinear discrimination.
- Problem solved with gradient ascent in the Stiefel manifold $\mathcal{S}$.
- Gradient computed using automatic differentiation of Sinkhorn algorithm.

Finding the (Monge) mapping

Mapping with optimal transport



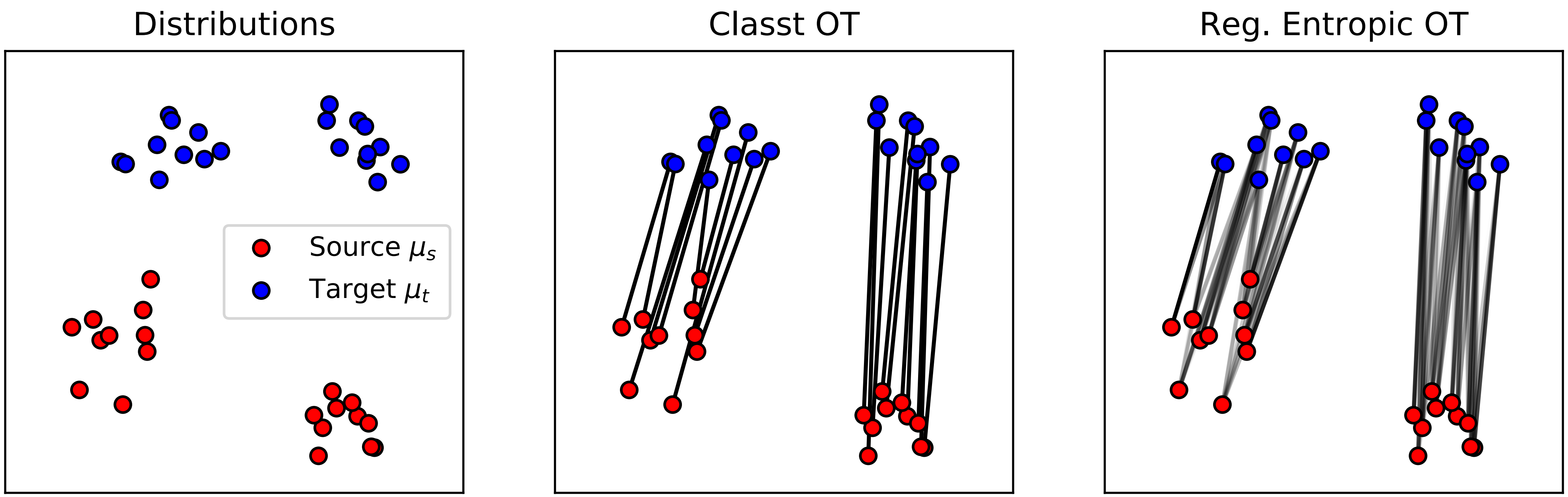
Mapping estimation

- Mapping do not exist in general between empirical distributions.
- Barycentric mapping [Ferradans et al., 2014].
- Smooth mapping estimation [Perrot et al., 2016, Seguy et al., 2017].

Why map ?

- Sensible displacement to align distributions.
- Color adaptation in image [Ferradans et al., 2014].
- Domain adaptation and transfer learning [Courty et al., 2016].

Transporting the discrete samples

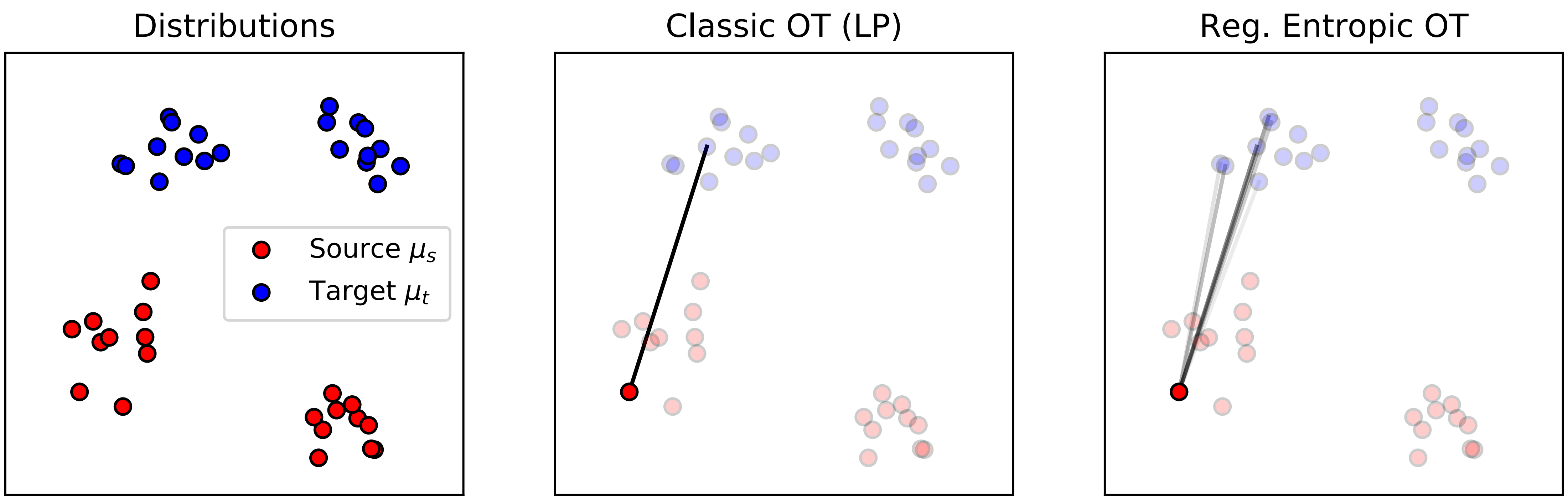


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \arg\min_{\mathbf{x}} \sum_j \gamma_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{1}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

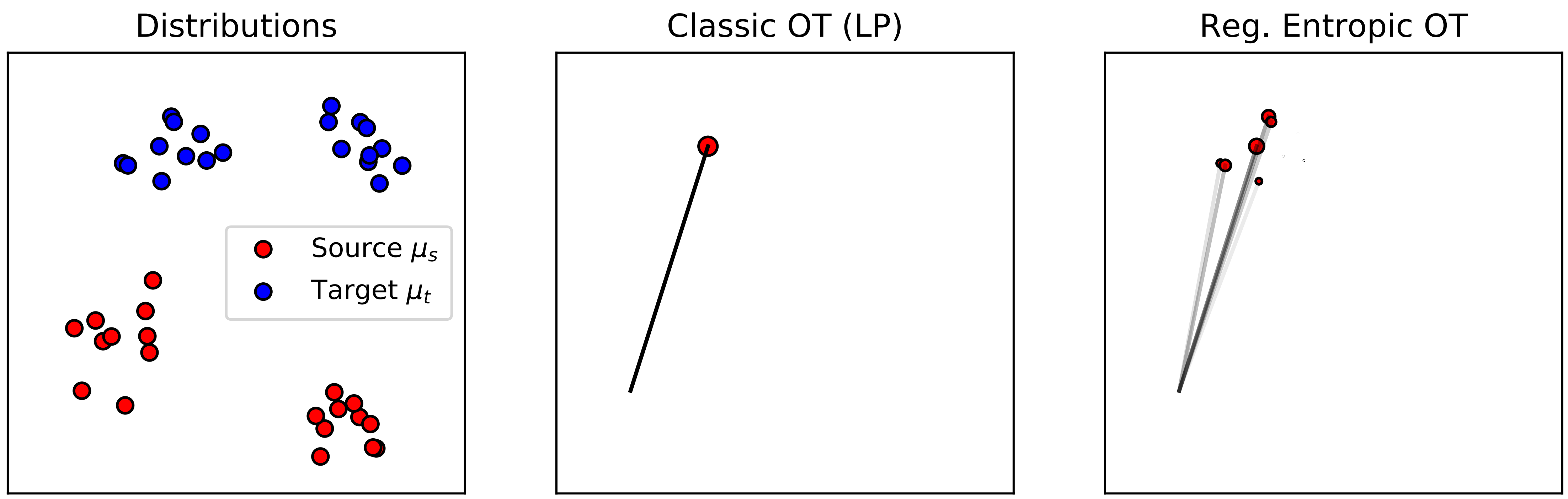


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \arg\min_{\mathbf{x}} \quad \sum_j \gamma_0(i,j)\|\mathbf{x} - \mathbf{x}_j^t\|^2. \quad (1)$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

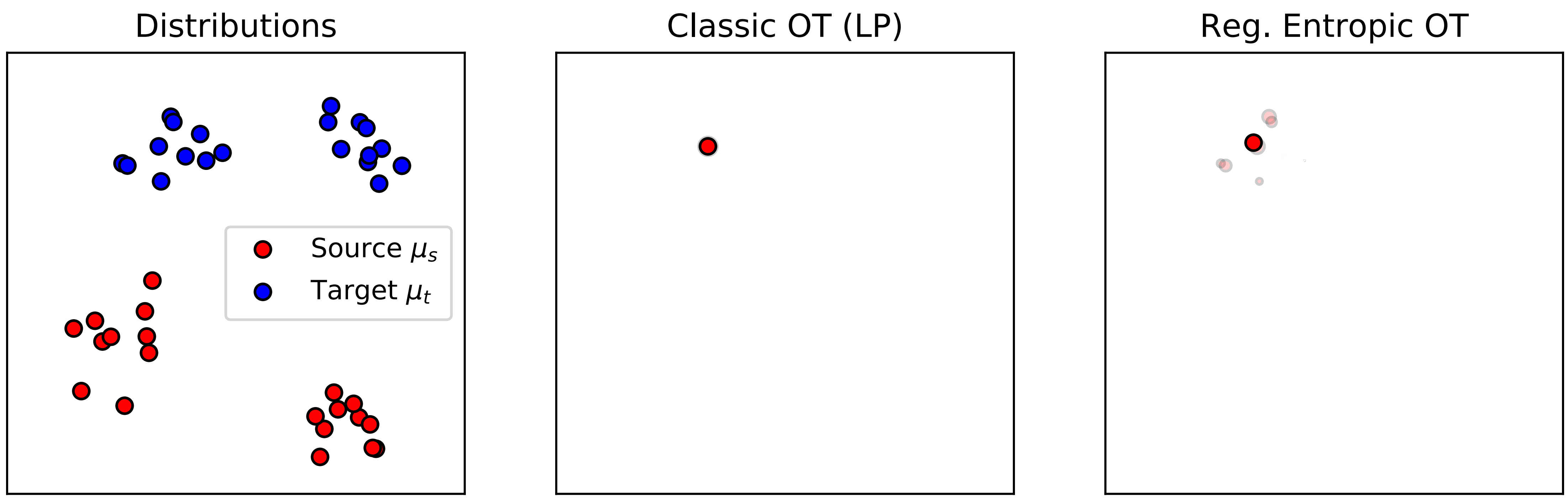


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \tag{1}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

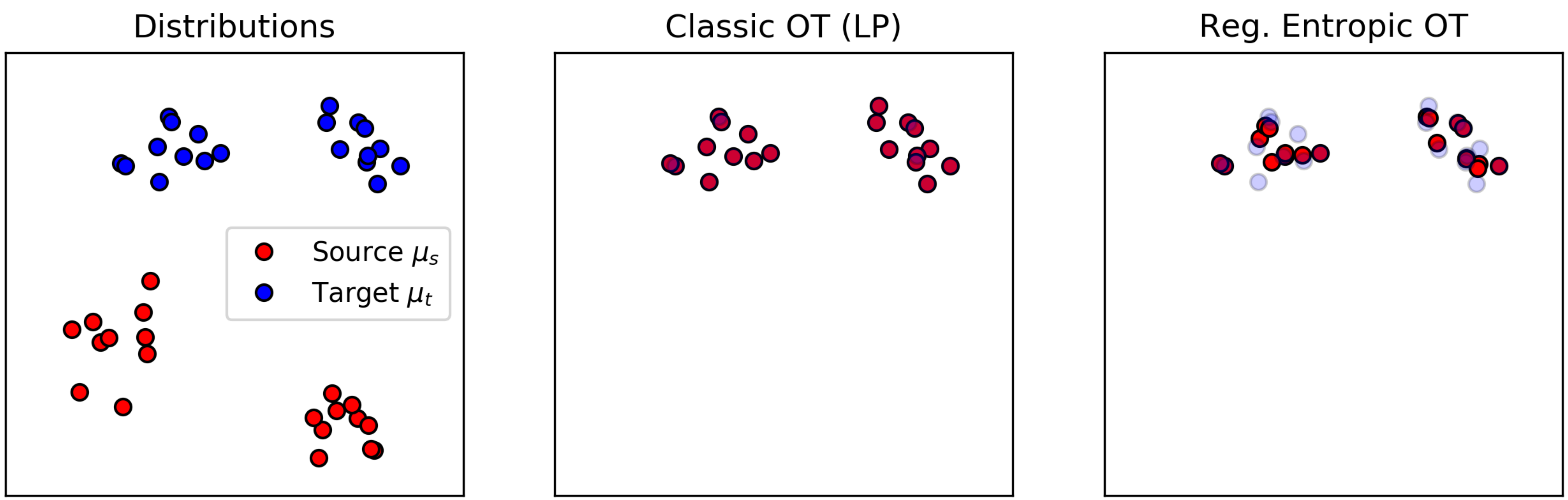


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \tag{1}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

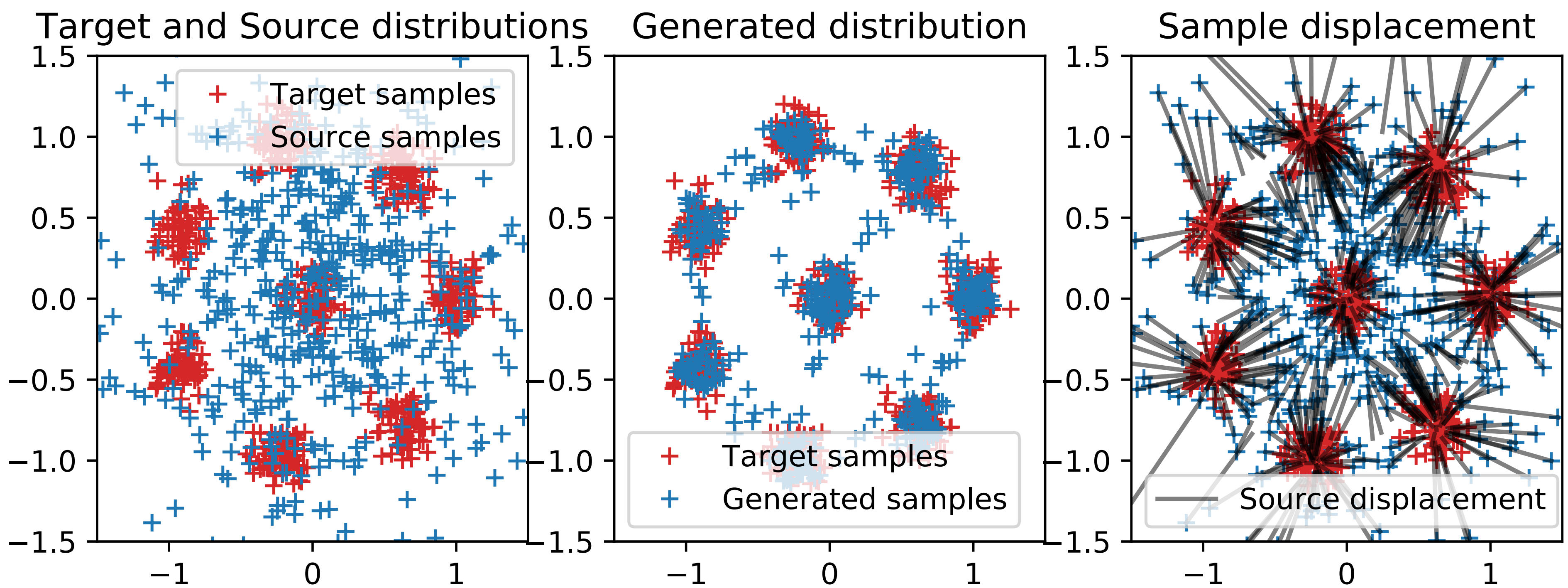


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \quad (1)$$

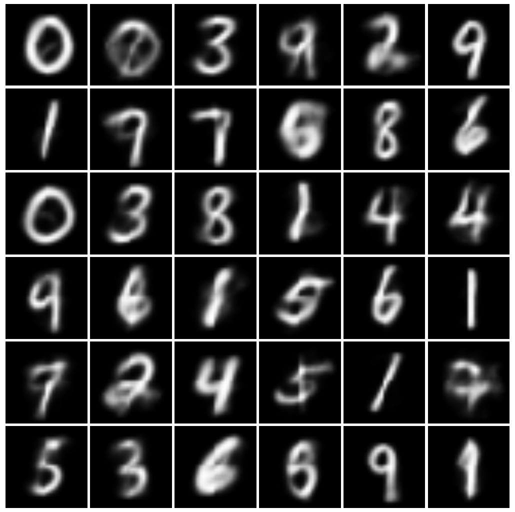
- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Large scale optimal transport and mapping estimation

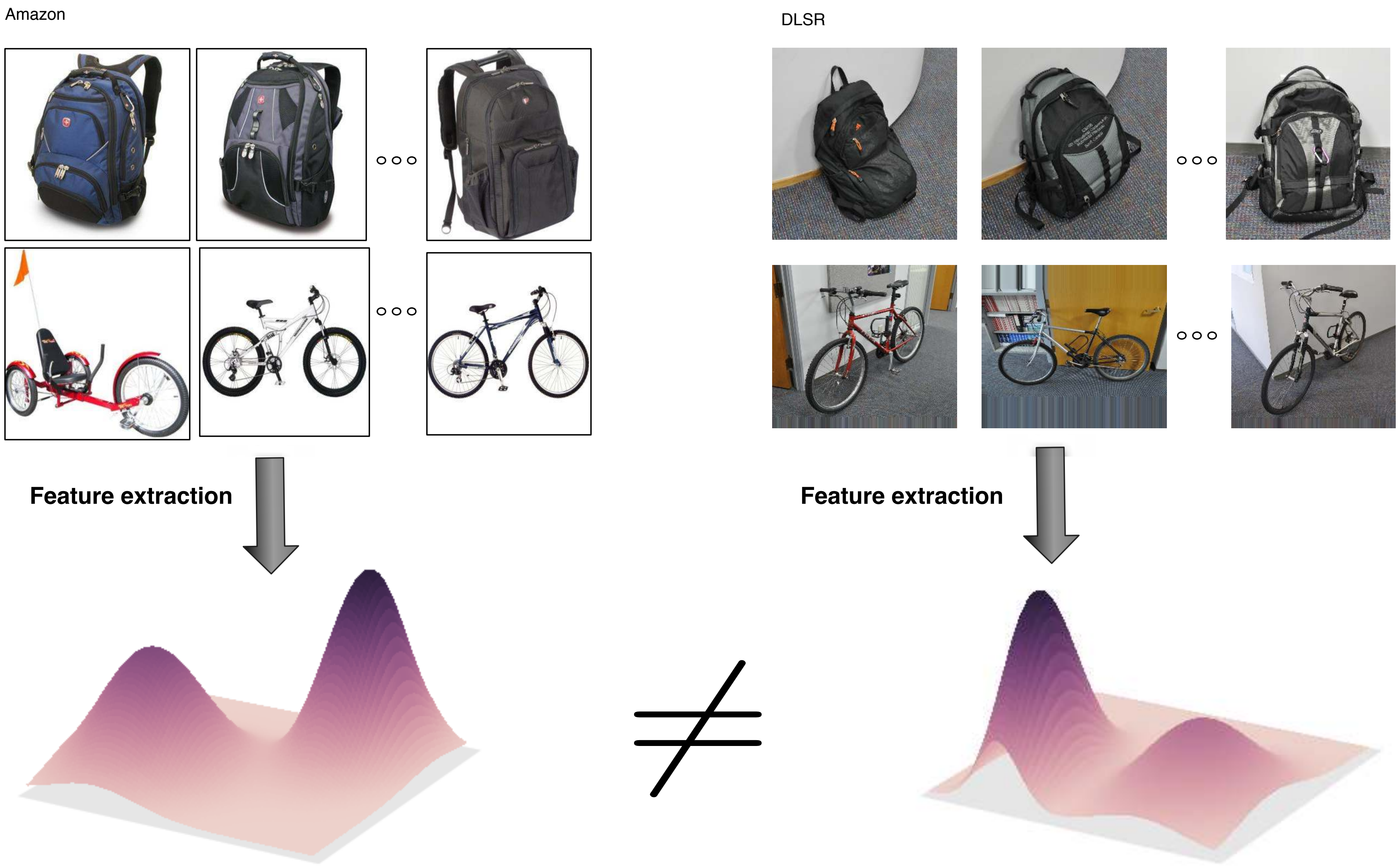


Large scale mapping estimation [Seguy et al., 2017]

- 2-step procedure:
  1. Stochastic estimation of regularized $\hat{\gamma}$.
  2. Stochastic estimation of f with a neural
- OT solved with Stochastic Gradient Ascent in the dual.
- Convergence to the true OT and mapping for small regularization.



Domain Adaptation problem

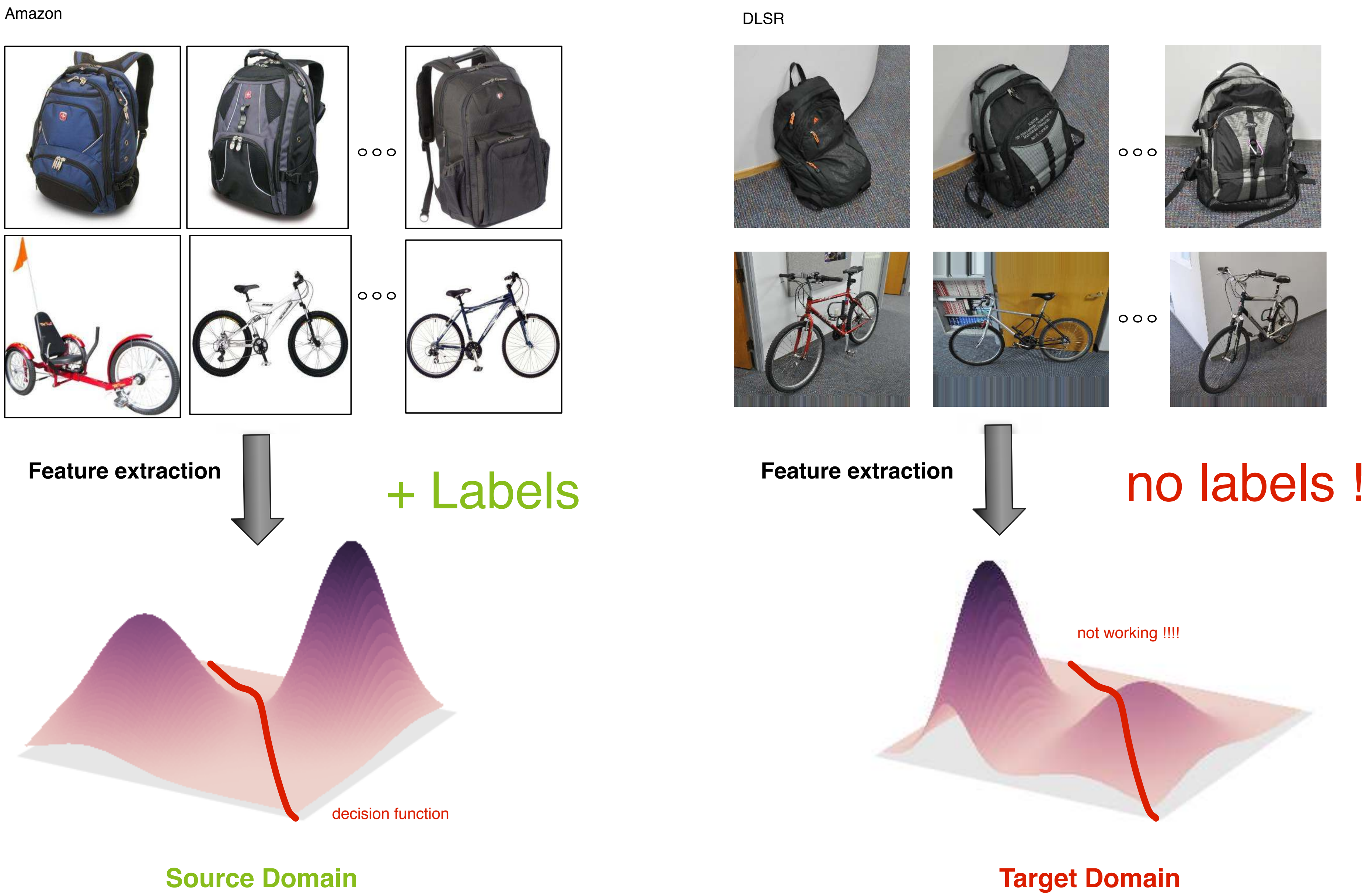


Probability Distribution Functions over the domains

Our context

- Classification problem with data coming from different sources (domains).
- Distributions are different but related.

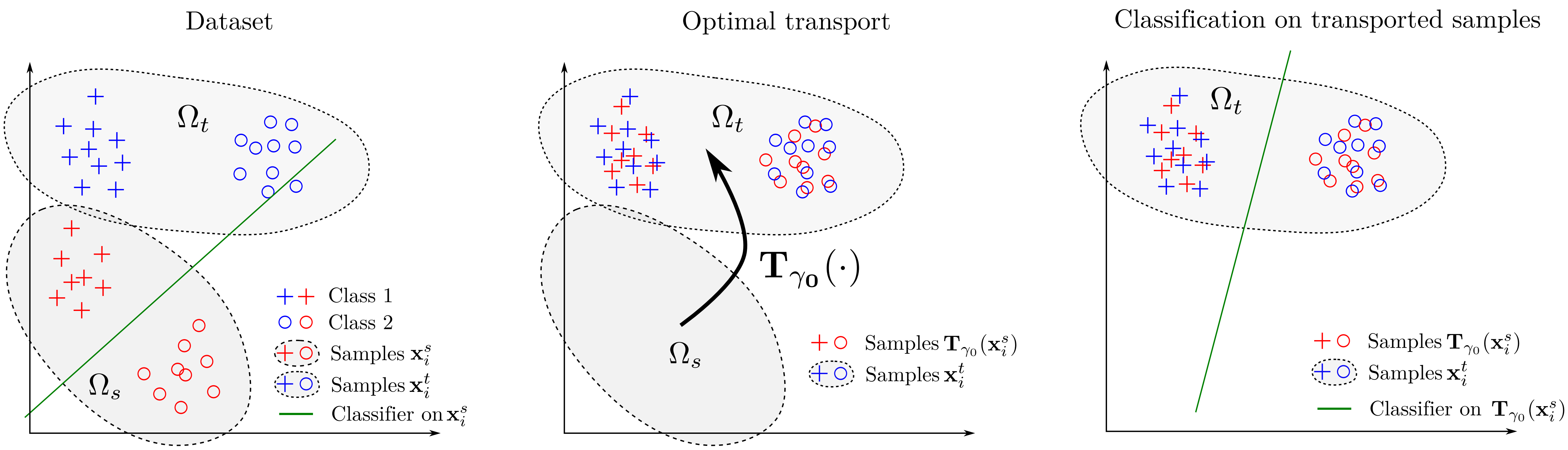
Unsupervised domain adaptation problem



Problems

- Labels only available in the **source domain**, and classification is conducted in the **target domain**.
- Classifier trained on the source domain data performs badly in the target domain

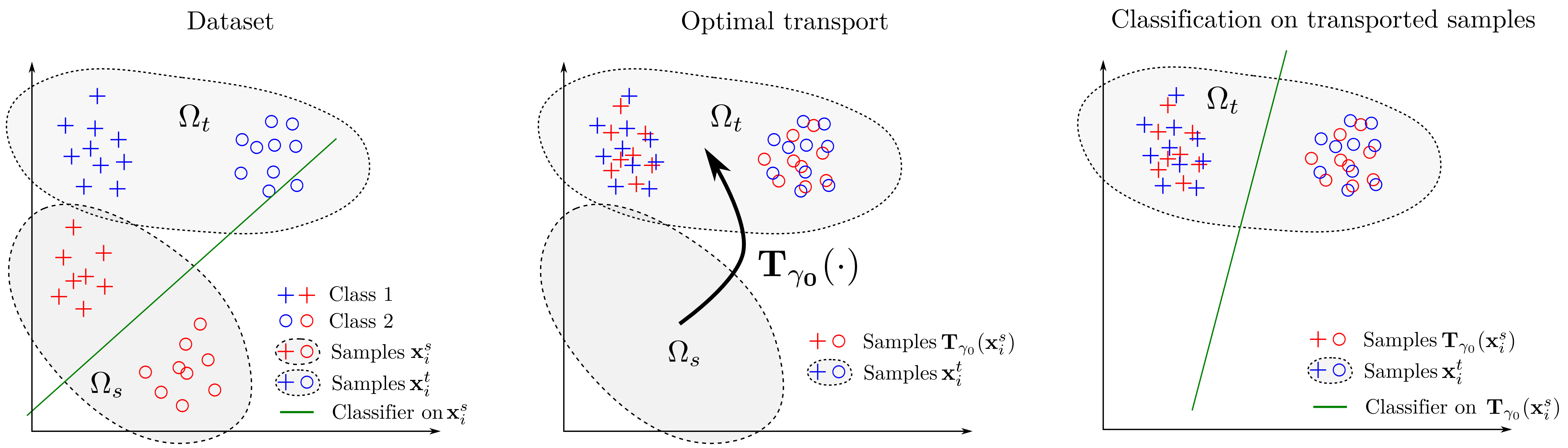
OT for domain adaptation : Step 1



Step 1 : Estimate optimal transport between distributions.

- Choose the ground metric (squared euclidean in our experiments).
- Using regularization allows
  - Large scale and regular OT with entropic regularization [Cuturi, 2013].
  - Class labels in the transport with group lasso [Courty et al., 2016].
- Efficient optimization based on Bregman projections [Benamou et al., 2015] and
  - Majoration minimization for non-convex group lasso.
  - Generalized Conditionnal gradient for general regularization (cvx. lasso, Laplacian).

OT for domain adaptation : Steps 2 & 3



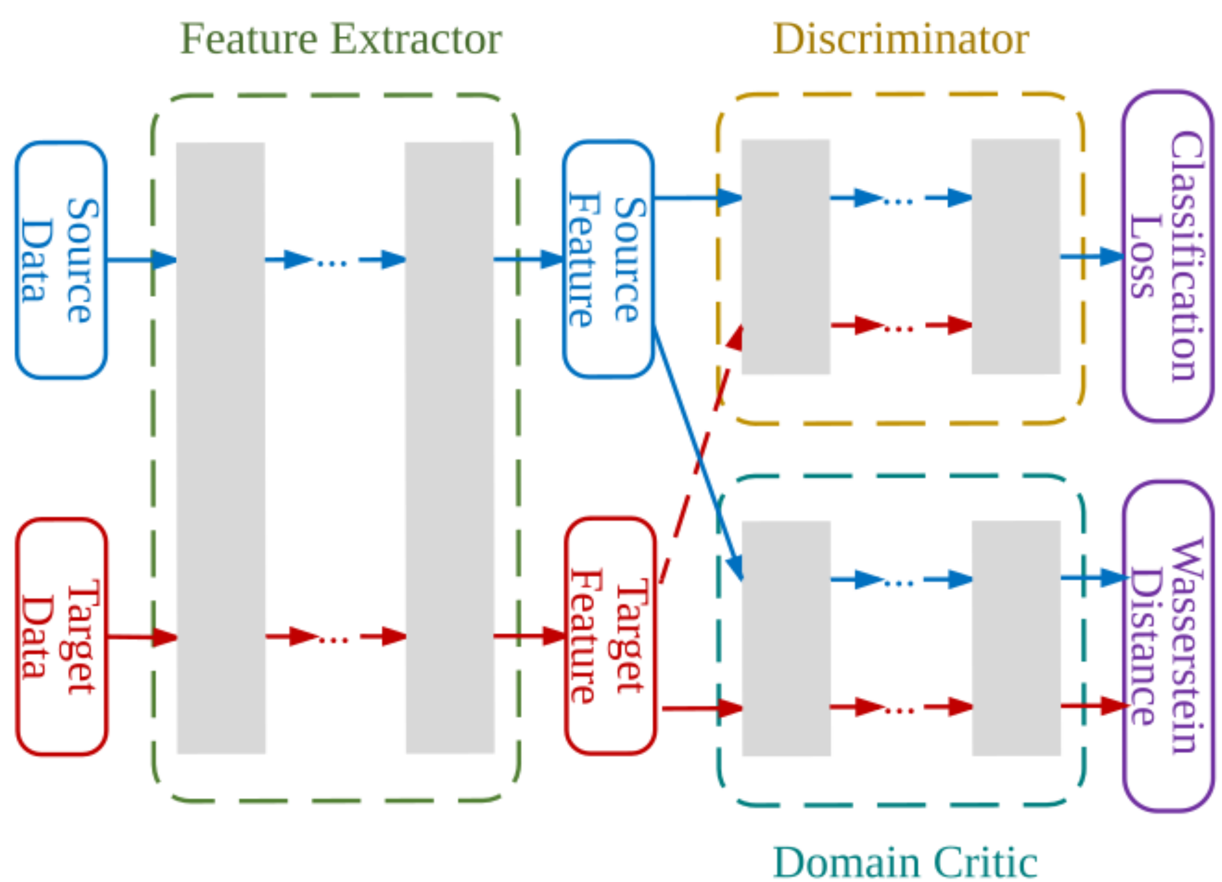
Step 2 : Transport the training samples onto the target distribution.

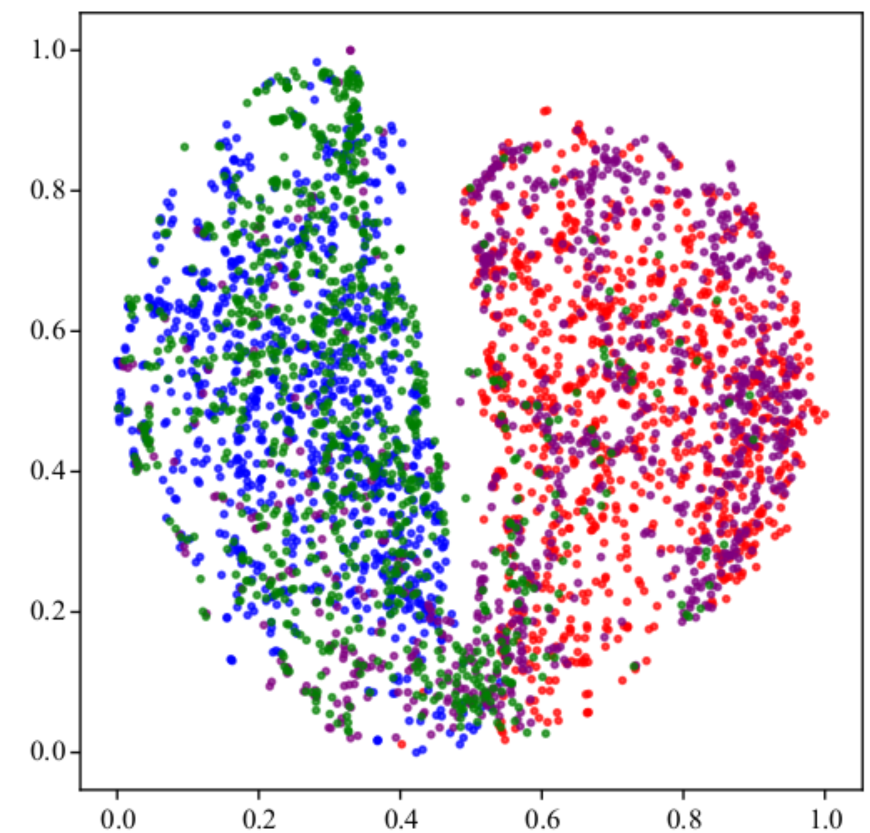
- The mass of each source sample is spread onto the target samples (line of γ_0).
- Transport using barycentric mapping [Ferradans et al., 2014].
- The mapping can be estimated for out of sample prediction [Perrot et al., 2016, Seguy et al., 2017].

Step 3 : Learn a classifier on the transported training samples

- Transported sample keep their labels.
- Classic ML problem when samples are well transported.

Domain adaptation with Wasserstein distance





(d) t-SNE of WDGRL features

Domain adaptation for deep learning [Shen et al., 2018]

- Modern DA aim at aligning source and target in the deep representation : DANN [Ganin et al., 2016], MMD [Tzeng et al., 2014], CORAL [Sun and Saenko, 2016].
- Wasserstein distance used as objective for the adaptation [Shen et al., 2018].

Joint Distribution Optimal Transport for DA

Learning with JDOT [Courty et al., 2017]

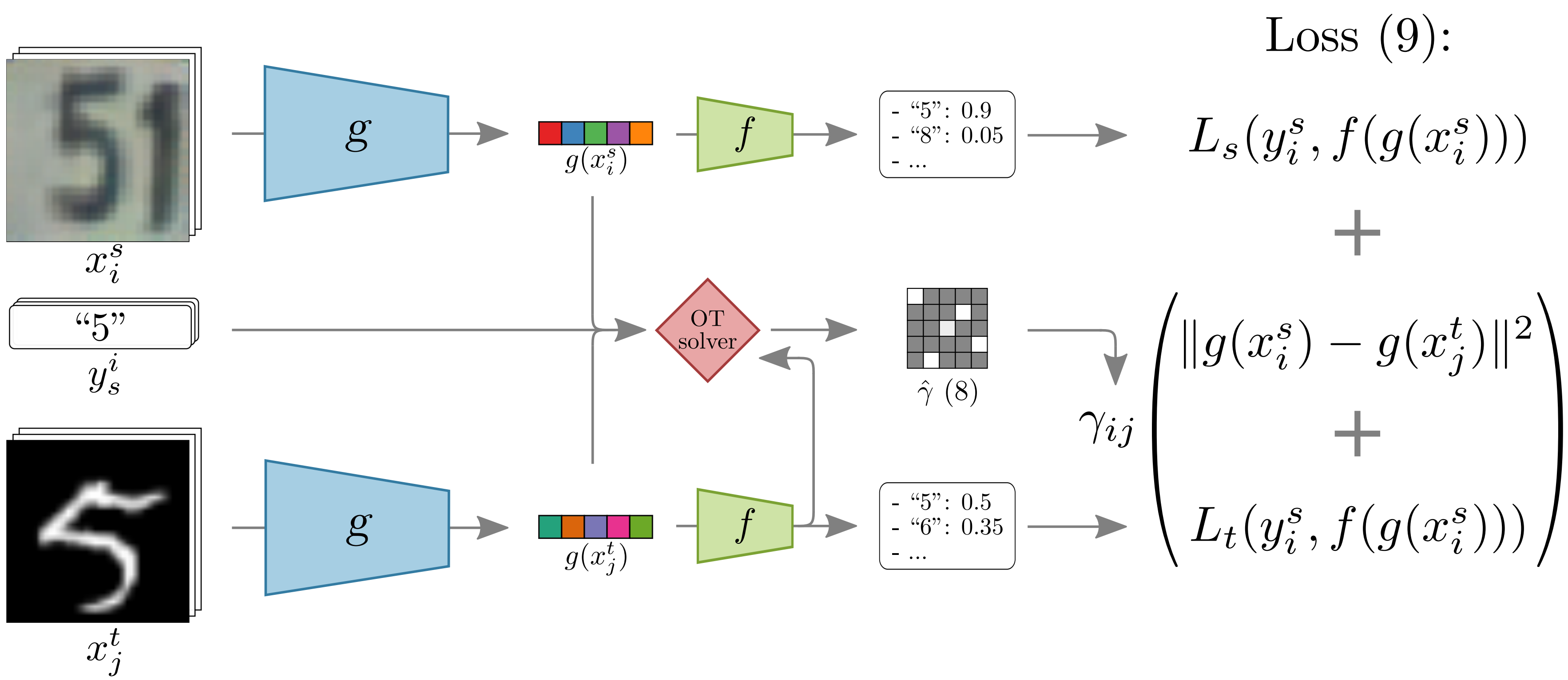
$$\min_f \quad \left\{ W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}_t}^f) = \inf_{\boldsymbol{\gamma}\in\Pi} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, y_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t))\boldsymbol{\gamma}_{ij} \right\} \tag{5}$$

- $\hat{\mathcal{P}_t}^f = \frac{1}{N_t}\sum_{i=1}^{N_t} \delta_{\mathbf{x}_i^t, f\mathbf{x}_i^t}$ is the proxy joint feature/label distribution.
- Π is the transport polytope, $\hat{\mathcal{P}}_s$ the empirical source distribution.
- $\mathcal{D}(\mathbf{x}_i^s, y_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) = \alpha\|\mathbf{x}_i^s - \mathbf{x}_j^t\|^2 + \mathcal{L}(y_i^s, f(\mathbf{x}_j^t))$ with $\alpha > 0$.
- We search for the predictor f that better align the joint distributions.
- JDOT can be seen as minimizing a generalization bound.

Optimizing JDOT

- Can be solved by block coordinate descent $(f, \boldsymbol{\gamma})$ [Courty et al., 2017].
- Solving with fixed f is classical OT.
- Solving with fixed $\boldsymbol{\gamma}$ is weighted empirical loss minimization.

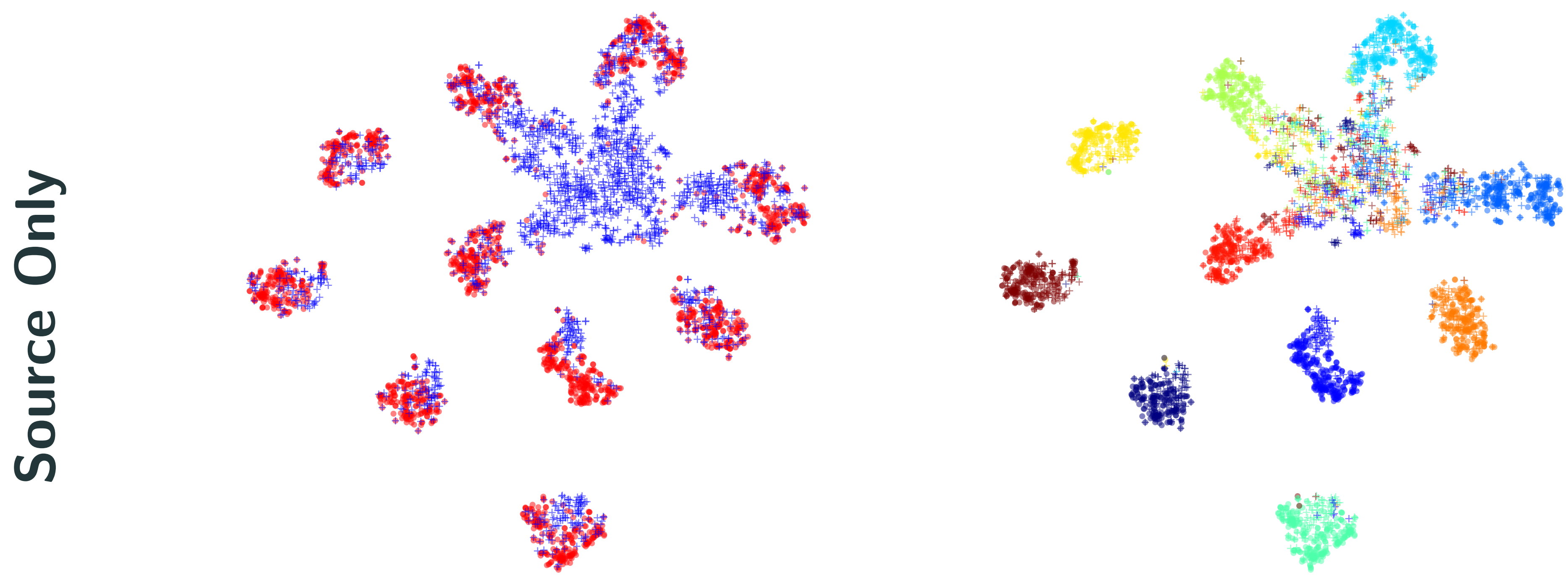
JDOT for large scale deep learning



DeepJDOT [Damodaran et al., 2018]

- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.
- Use minibatch to estimate OT and update g, f at each iterations.
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST→MNIST-M).

JDOT for large scale deep learning



DeepJDOT [Damodaran et al., 2018]

- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.
- Use minibatch to estimate OT and update g, f at each iterations.
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).

JDOT for large scale deep learning

DeepJDOT

DeepJDOT [Damodaran et al., 2018]

- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.
- Use minibatch to estimate OT and update g, f at each iterations.
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).

3. Conclusion

Conclusion

- A powerful tool, well theoretically grounded, for manipulating distributions in ML
- Despite its initial computational complexity, a lot of applications, even in large scale/deep learning settings
- Uncovered aspects (in this presentation): unbalanced OT, Gromov-Wasserstein (working with structured data), and many more !
- Codes available !

POT (PYTHON OPTIMAL TRANSPORT TOOLBOX)

README.md

POT: Python Optimal Transport

pypi package 0.4.0 | build passing | docs passing

This open source Python library provide several solvers for optimization problems related to Optimal Transport for signal, image processing and machine learning.

It provides the following solvers:

- OT solver for the linear program/ Earth Movers Distance [1].
- Entropic regularization OT solver with Sinkhorn Knopp Algorithm [2] and stabilized version [9][10] with optional GPU implementation (required cudamat).
- Bregman projections for Wasserstein barycenter [3] and unmixing [4].
- Optimal transport for domain adaptation with group lasso regularization [5]
- Conditional gradient [6] and Generalized conditional gradient for regularized OT [7].
- Joint OT matrix and mapping estimation [8].
- Wasserstein Discriminant Analysis [11] (requires autograd + pymanopt).
- Gromov-Wasserstein distances and barycenters [12]

Some demonstrations (both in Python and Jupyter Notebook format) are available in the examples folder.

Installation

The library has been tested on Linux, MacOSX and Windows. It requires a C++ compiler for using the EMD solver and relies on the following Python modules:

- Numpy (>=1.11)

References i

Arjovsky, M., Chintala, S., and Bottou, L. (2017).
Wasserstein gan.
arXiv preprint arXiv:1701.07875.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).
Iterative Bregman projections for regularized transportation problems.
SISC.

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. (2017).
Joint distribution optimal transportation for domain adaptation.
In *Neural Information Processing Systems (NIPS)*.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016).
Optimal transport for domain adaptation.
Pattern Analysis and Machine Intelligence, IEEE Transactions on.

Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transportation.
In *Neural Information Processing Systems (NIPS)*, pages 2292–2300.

Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018).
Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation.

Emiya, V., Badeau, R., and David, B. (2010).
Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle.
IEEE Transactions on Audio, Speech, and Language Processing, 18(6):1643–1654.

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014).
Regularized discrete optimal transport.
SIAM Journal on Imaging Sciences, 7(3).

Flamary, R., Cuturi, M., Courty, N., and Rakotomamonjy, A. (2016a).
Wasserstein discriminant analysis.
arXiv preprint arXiv:1608.08063.

Flamary, R., Fevotte, C., Courty, N., and Emyia, V. (2016b).
Optimal spectral transportation with application to music transcription.
In *Neural Information Processing Systems (NIPS)*.

Frogner, C., Zhang, C., Mobahi, H., Araya, M., and Poggio, T. A. (2015).
Learning with a wasserstein loss.
In *Advances in Neural Information Processing Systems*, pages 2053–2061.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016).
Domain-adversarial training of neural networks.
Journal of Machine Learning Research, 17(59):1–35.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
Generative adversarial nets.
In *Advances in neural information processing systems*, pages 2672–2680.

References iv

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017).
Improved training of wasserstein gans.
In *Advances in Neural Information Processing Systems*, pages 5769–5779.

Montavon, G., Müller, K.-R., and Cuturi, M. (2016).
Wasserstein training of restricted boltzmann machines.
In *Advances in Neural Information Processing Systems*, pages 3718–3726.

Pérez, P., Gangnet, M., and Blake, A. (2003).
Poisson image editing.
ACM Trans. on Graphics, 22(3).

Perrot, M., Courty, N., Flamary, R., and Habrard, A. (2016).
Mapping estimation for discrete optimal transport.
In *Neural Information Processing Systems (NIPS)*.

Radford, A., Metz, L., and Chintala, S. (2015).
Unsupervised representation learning with deep convolutional generative adversarial networks.
arXiv preprint arXiv:1511.06434.

Rolet, A., Cuturi, M., and Peyré, G. (2016).
Fast dictionary learning with a smoothed wasserstein loss.
In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 630–638.

Sandler, R. and Lindenbaum, M. (2011).
Nonnegative matrix factorization with earth mover's distance metric for image analysis.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(8):1590–1602.

Schmitz, M. A., Heitz, M., Bonneel, N., Mboula, F. M. N., Coeurjolly, D., Cuturi, M., Peyré, G., and Starck, J.-L. (2017).
Wasserstein dictionary learning: Optimal transport-based unsupervised non-linear dictionary learning.
arXiv preprint arXiv:1708.01955.

Seguy, V., Bhushan Damodaran, B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. (2017).
Large-scale optimal transport and mapping estimation.

Shen, J., Qu, Y., Zhang, W., and Yu, Y. (2018).
Wasserstein distance guided representation learning for domain adaptation.
In *AAAI Conference on Artificial Intelligence*.

Sun, B. and Saenko, K. (2016).
Deep CORAL: Correlation Alignment for Deep Domain Adaptation, pages 443–450.
Springer International Publishing, Cham.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014).
Deep domain confusion: Maximizing for domain invariance.
arXiv preprint arXiv:1412.3474.

Zen, G., Ricci, E., and Sebe, N. (2014).
Simultaneous ground metric learning and matrix factorization with earth mover's distance.
In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3690–3695.

Zhao, J., Mathieu, M., and LeCun, Y. (2016).
Energy-based generative adversarial network.
arXiv preprint arXiv:1609.03126.